

Conversion de dates en Java

1. Le type `java.time.LocalDate`

Depuis la version 8 de Java, le type conseillé pour enregistrer les dates est `LocalDate` du paquetage `java.time`.

1.1. Conversion `LocalDate` vers `String` :

utiliser un objet de type `DateTimeFormatter` :

```
LocalDate uneDate = LocalDate.of(2018,4,17); //jeu d'essai
DateTimeFormatter dtf = DateTimeFormatter.ofPattern("yyyy-MM-dd");
String maDateStr = dtf.format(uneDate);
```

1.2. Conversion `String` vers `LocalDate` :

utiliser un objet de type `DateTimeFormatter` :

```
String uneDateStr = "17/04/2018"; // jeu d'essai
DateTimeFormatter dtf = DateTimeFormatter.ofPattern("dd/MM/yyyy");
LocalDate d = LocalDate.parse(uneDateStr, dtf);
```

2. Le type `java.sql.Date`

Lorsqu'il lit une donnée issue d'un champ du type SQL DATE, JDBC instancie un objet de type `java.sql.Date` (sous-type de `java.util.Date`) : `java.sql.Date ddn = rs.getDate("DATEDENAISSANCE") ;`

2.1. Convertir une date de type « `java.sql.Date` » en « `LocalDate` »

```
LocalDate uneDate = rs.getDate("DATEDENAISSANCE").toLocalDate() ;
```

2.2. Convertir une date de type « `LocalDate` » en « `java.sql.Date` »

```
LocalDate uneDate = LocalDate.of(1999,1,15) ;
java.sql.Date ddn = java.sql.Date.valueOf(uneDate) ;
```