

Temporal- Dates

@Temporal Annotation in Spring (JPA) [↗](#)

- **Where it belongs:**

It's actually part of **JPA** (`javax.persistence.Temporal`), not Spring itself. But in **Spring Data JPA** projects, you often see it because Spring uses JPA under the hood.

- **Purpose:**

`@Temporal` tells JPA **how** to map a Java `java.util.Date` or `java.util.Calendar` field into a SQL database column — because in Java, a `Date` contains **both date and time**, but in SQL you might want to store just the **date**, just the **time**, or **both**.

Syntax: [↗](#)

```
1 @Temporal(TemporalType.DATE)
2 private Date birthDate;
3
```

TemporalType Options: [↗](#)

TemporalType	Meaning	Example
DATE	Only the date part is stored. (no time)	2025-04-29
TIME	Only the time part is stored. (no date)	12:30:45
TIMESTAMP	Both date and time are stored.	2025-04-29 12:30:45

Example 1: Storing Only Date [↗](#)

```
1 @Entity
2 public class Person {
3
4     @Id
5     private Long id;
```

```

6
7     @Temporal(TemporalType.DATE)
8     private Date birthDate;
9 }
10

```

- Here `birthDate` will be saved in the database like `2025-04-29` (without any time part).

Example 2: Storing Full Date + Time [↗](#)

```

1 @Entity
2 public class Event {
3
4     @Id
5     private Long id;
6
7     @Temporal(TemporalType.TIMESTAMP)
8     private Date eventTimestamp;
9 }
10

```

- Now `eventTimestamp` will store both date and time like `2025-04-29 14:55:00`.

Important Notes: [↗](#)

- **Required only for** `java.util.Date` **and** `Calendar`.
Newer classes like `LocalDate`, `LocalDateTime` (from Java 8's `java.time` package) **don't need** `@Temporal`.
- If you use `LocalDate`, `LocalDateTime`, or `Instant`, Spring Data JPA understands them automatically without `@Temporal`.
- If you **forget** to use `@Temporal` with `Date`, you might get unexpected results (e.g., wrong format in the database).

Why is it needed? [↗](#)

Because **Java Date** is ambiguous — it doesn't clearly separate date and time — but **SQL** types are **clear** (`DATE`, `TIME`, `TIMESTAMP`), so we must **tell JPA how to store it**.

Quick Visual: [↗](#)

```

1 Java's Date
2 |
3 |--@Temporal(DATE) --> SQL Date (Only 2025-04-29)
4 |--@Temporal(TIME) --> SQL Time (Only 12:30:45)
5 |--@Temporal(TIMESTAMP) --> SQL Timestamp (Both date + time)
6

```

