FEEL - Temporals

What I Will Learn

This course gives a detailed hands-on experience on 🔗 FEEL **Temporal** data types, expressions and functions used in Camunda.

You will practice using the **FEEL Evaluator**, an interface similar to the FEEL Playground Online, which will allow you to evaluate, lesson by lesson, the FEEL expressions you are learning.

What Will I Learn?

At the end of this course you will be able to:

- Understand the basic temporal data types:
    - date
    - time
    - date and time
    - day-time duration
    - year-month duration
- Operate with various temporal expressions:
    - **Addition**
    - **Subtraction**
    - **Multiplication**
    - **Division**
- Recognize most common temporal attributes:
    - year, years
    - month, months
    - day, days
    - weekday
    - hour, hours
    - minute, minutes

  - second, seconds
  - time offset
  - timezone
- Use various temporal functions:
  - now()
  - today()
  - day of week()
  - day of year()
  - week of year()
  - month of year()
  - last day of month()

Data Types

Date

The **date** data type in the 🔗 DMN FEEL is used to represent calendar dates. This data type is essential for decision logic that involves date comparisons, calculations, and formatting. All these topics will be cover along the course.

Basic Concepts

Representation: A **date** in FEEL is typically represented in the format yyyy-MM-dd. For example, January 1, 2025, is written as **2025-01-01**.

Creation: Dates are created using the date function.

Examples

Creating Dates

To create a date in FEEL, you use the date function:

```
date("2024-06-10")
```

Also, you can create a date using the "**@**" prefix before the date:

```
@"2024-06-10"
```

Info

Both expressions are equivalent, and create a date representing June 10, 2024

They both will return this result:

"2024-06-10"

Time

The **time** data type in the 🔗 DMN FEEL is used to represent specific times of the day. This data type is essential for decision logic that involves time comparisons, calculations, and formatting. All these topics will be cover along the course.

Basic Concepts

Representation: A **time** in FEEL is typically represented in one of these formats: HH:mm:ss, HH:mm:ss+/-HH:mm, and HH:mm:ss@ZoneId. For example, 14:30 (2:30 PM) could be written as **14:30:00**.

Creation: Times are created using the time function.

Examples

Creating Times

To create a time in FEEL, you use the time function:

time("14:30:00")

This expression creates a time representing 2:30 PM.

It will return this result:

"14:30:00"

Different formats could be used to create times:

time("13:30")

time("11:45:30+02:00")

time("10:31:10@Europe/Paris")

Also, you can create a time using the "**@**" prefix before the time:

@"11:45:30"

@"13:30"

@"11:45:30+02:00"

@"10:31:10@Europe/Paris"

Date and Time

The **Date and Time** data type in the 🔗 DMN FEEL is used to represent specific points in time, combining both date and time information. This data type is essential for decision logic that involves datetime comparisons, calculations, and formatting. All these topics will be cover along the course.

Basic Concepts

Representation: A **Date and Time** in FEEL is typically represented in one of these formats: yyyy-MM-dd'T'HH:mm:ss, yyyy-MM-dd'T'HH:mm:ss+/-HH:mm, and yyyy-MM-dd'T'HH:mm:ss@ZoneId. For example, January 1, 2025, at 14:30 (2:30 PM) is written as **2025-01-01T14:30:00**.

Creation: Datetimes are created using the date and time function.

Examples

Creating Datetimes

To create a date and time in FEEL, you use the date and time function:

date and time("2025-01-01T14:30:00")

This expression creates a datetime representing January 1, 2025, at 2:30 PM.

It will return this result:

"2025-01-01T14:30:00"

Different formats could be used to create datetimes:

date and time("2025-06-10T10:31:10+01:00")

date and time("2025-06-10T10:31:10@Europe/Paris")

Also, you can create a date and time using the "**@**" prefix before the date and time:

@"2025-06-10T10:31:10"

@"2025-06-10T10:31:10+01:00"

@"2025-06-10T10:31:10@Europe/Paris"

Duration

The **duration** data type in the 🔗 DMN FEEL is used to represent periods of time. This data type is essential for decision logic that involves adding to or subtracting from dates, times, or datetimes, as well as for comparing durations. All these topics will be cover along the course.

Basic Concepts

Representation: **Durations** in FEEL are represented in the ISO 8601 format. There are two types of durations:

- **Year-Month** duration: Represented as PnYnM (e.g., P2Y3M for 2 years and 3 months).
- **Day-Time** duration: Represented as PnDTnHnMnS (e.g., P1DT2H3M4S for 1 day, 2 hours, 3 minutes, and 4 seconds).

Info

- **P** represents **P**eriod
- **T** represents **T**ime
- Use "**P**" + **NUMBER** + "**D**" for days.
- Use "**PT**" + **NUMBER** + "**H**" for hours, or "**M**" for minutes, or "**S**" for seconds.
- Usually, the result will try to be returned in "**PT**" value.

Creation: Durations are created using the duration function.

Examples

Creating Day-Time Durations

To create a day-time duration in FEEL, you use the duration function:

duration("P1DT6H3M")

This expression creates a duration representing 1 day, 6 hours and 3 minutes.

It will return this result:

"PT30H3M"

Info

The result is returned in "**P**eriod **T**ime" value.

Different formats could be used to create date-time durations:

duration("P4D")

duration("PT2H")

duration("PT30M")

duration("P1DT6H")

Also, you can create a date-time duration using the "**@**" prefix before the duration:

@"P4D"

@"PT2H"

@"PT30M"

@"P1DT6H"

Creating Year-Month Durations

To create a year-month duration in FEEL, you use the duration function:

duration("P1Y6M")

This expression creates a duration representing 1 year, and 6 months.

It will return this result:

"P1Y6M"

Different formats could be used to create year-month durations:

duration("P2Y")

duration("P6M")

Also, you can create a year-month duration using the "**@**" prefix before the duration:

@"P2Y"

@"P6M"

@"P1Y6M"

Expressions

Addition

In this lesson, we will explore how to perform addition operations between different data types in DMN FEEL. We will cover:

- Adding durations to dates
- Adding durations to times

- Adding durations to date and time
- Adding durations to durations

By practicing these operations, you will become proficient in handling various date, time, and duration manipulations in your decision models using FEEL.

Date

Info

DATE + DURATION = DATE

When you add a duration to a date, you adjust the date by the specified period.

Adding a day-time duration to a date

date("2025-01-01") + duration("P10DT24H2M")

Result: "2025-01-12"

Adding a year-month duration to a date

date("2025-01-01") + duration("P2Y3M")

Result: "2027-04-01"

Time

Info

TIME + DURATION = TIME

When you add a duration to a time, you adjust the time by the specified period.

Adding hours and minutes to a time

To add hours and minutes, you can combine different duration parts. For example, adding 1 hour and 45 minutes:

time("14:30:00") + duration("PT1H45M")

Result: "16:15:00"

Date and time

Info

DATE-TIME + DURATION = DATE-TIME

When you add a duration to a date and time, you adjust both the date and the time by the specified period.

Adding a year-month duration to a datetime

date and time("2025-01-01T14:30:00") + duration("P1Y3M")

Result: "2026-04-01T14:30:00"

Adding a day-time duration to a datetime

date and time("2025-01-01T14:30:00") + duration("P3DT4H5M6S")

Result: "2025-01-04T18:35:06"

Duration

Info

DURATION + DURATION = DURATION

Adding duration to duration

The two arguments of the expression must have a day-time or a year-month duration.

duration("P2D") + duration("P5D")

Result: "PT168H"

Null result

The addition of these different data types will return null:

- DATE + DATE = NULL
- TIME + TIME = NULL
- TIME + DATE = NULL
- DURATION(YM) + DURATION(DT) = NULL

Subtraction

In this lesson, we will explore how to perform subtraction operations between different data types in DMN FEEL. We will cover:

- Subtracting dates from dates
- Subtracting durations from dates
- Subtracting durations from times
- Subtracting durations from date and time
- Subtracting durations from durations

By practicing these operations, you will become proficient in handling various date, time, and duration manipulations in your decision models using FEEL.

Date

Info

DATE - DURATION = DATE

When you subtract a duration from a date, you adjust the date by the specified period.

Subtracting a day-time duration from a date

date("2025-01-01") - duration("P10DT24H2M")

Result: "2024-12-20"

Subtracting a year-month duration from a date

date("2025-01-01") - duration("P2Y3M")

Result: "2022-10-01"

Info

DATE - DATE = DURATION(DT)

When you subtract a date from a date, you adjust the dates by the specified period.

Subtracting a date from a date

date("2035-01-02") - date("2035-01-01")

Result: "PT24H"

Info

The result is a day-time duration

Time

Info

TIME - DURATION = TIME

When you subtract a duration from a time, you adjust the time by the specified period.

Subtracting hours and minutes from a time

To subtract hours and minutes, you can combine different duration parts. For example, subtracting 1 hour and 45 minutes:

time("14:30:00") - duration("PT1H45M")

Result: "12:45:00"

Also, you can use the time() function to subtract hours and minutes:

time("14:30:00") - time("16:30:00")

Result: "PT-2H"

Note

When the result is negative, it's indicated by a dash **"-"**.

You can use the abs() function to return the absolute value of a given duration:

abs(duration("-PT5H"))

Result: "PT5H"

Date and time

Info

DATE-TIME - DURATION = DATE-TIME

When you subtract a duration from a date and time, you adjust both the date and the time by the specified period.

Subtracting a year-month duration from a datetime

date and time("2025-01-01T14:30:00") - duration("P1Y3M")

Result: "2023-10-01T14:30:00"

Subtracting a day-time duration from a datetime

date and time("2025-01-01T14:30:00") - duration("P3DT4H5M6S")

Result: "2024-12-29T10:24:54"

Duration

Info

DURATION - DURATION = DURATION

Subtracting duration from duration

The two arguments of the expression must have a day-time or a year-month duration.

duration("P2D") - duration("P5D")

Result: "PT-72H"

Null result

The subtraction of these different data types will return null:

- TIME - DATE = NULL
- DURATION(YM) - DURATION(DT) = NULL

Multiplication

In this lesson, we will explore how to perform multiplication operations between different data types in DMN FEEL. We will cover:

- Multiplying **day-time** durations by a number
- Multiplying **year-month** durations by a number

By practicing these operations, you will become proficient in handling various duration manipulations in your decision models using FEEL.

Durations

Info

DURATION * NUMBER = DURATION

When you multiply a duration by a number, you adjust both the duration and the number.

Multiplying a year-month duration by a number

duration("P1Y3M") * 2

Result: "P2Y6M"

Multiplying a day-time duration by a number

duration("P1DT4H5M6S") * 3

Result: "PT84H15M18S"

Info

The result is returned in "**P**eriod **T**ime" value.

Null result

The multiplication of different data types will return null:

Division

In this lesson, we will explore how to perform division operations between different data types in DMN FEEL. We will cover:

- Dividing durations by a number
- Dividing **day-time** durations by **day-time** durations
- Dividing **year-month** durations by **year-month** durations

By practicing these operations, you will become proficient in handling various duration manipulations in your decision models using FEEL.

Durations

Info

DURATION / NUMBER = DURATION

When you divide a duration by a number, you adjust both the duration and the number.

Info

DURATION / DURATION = NUMBER

When you divide a duration by a duration, you adjust the duration of both operators. The expression requires that the two arguments have either a day-time or a year-month duration.

Dividing a year-month duration by a number

duration("P1Y") / 12

Result: "P1M"

Dividing a year-month duration by a year-month duration

duration("P1Y") / duration("P1M")

Result: 12

Dividing a day-time duration by a number

duration("P5D") / 5

Result: "PT24H"

Info

The result is returned in "**P**eriod **T**ime" value.

Dividing a day-time duration by a day-time duration

duration("P5D") / duration("P1D")

Result: 5

Null result

The division of different data types will return null:

Attributes

In this lesson, we will explore the attributes of the different data types in DMN FEEL. We will cover the attributes for:

- date
- time
- date and time
- day-time duration
- year-month duration

By practicing these attributes, you will become proficient in handling various date, time, and duration manipulations in your decision models using FEEL.

Date and Datetime

Data types date and date and time have these attributes:

- year as number
- month as number [1..12], where 1 is January
- day as number [1..31]
- weekday as number [1..7], where 1 is Monday

Getting the year of a date

date("2025-04-06").year

Result: 2025

Getting the month of a datetime

date and time("2025-04-06T08:00:00+02:00").month

Result: 4

Getting the day of a date

date("2025-04-06").day

Result: 6

Getting the weekday of a datetime

date and time("2025-04-06T08:00:00+02:00").weekday

Result: 7

Time and Datetime

Data types time and date and time have these attributes:

- hour as number [0..23]
- minute as number [0..59]
- second as number [0..59]
- time offset as duration offset corresponding to the timezone
- timezone as identifier

Getting the hour of a time

time("08:00:00").hour

Result: 8

Getting the minute of a datetime

date and time("2025-04-06T08:04:00+02:00").minute

Result: 4

Getting the second of a time

time("08:04:41").second

Result: 41

Getting the time offset of a datetime

date and time("2025-04-06T08:00:00+02:00").time offset

Result: "PT2H"

Getting the timezone of a datetime

@"2025-06-10T10:31:10@Europe/Madrid".timezone

Result: "Europe/Madrid"

Getting the timezone of a time

time("08:00:00@Europe/Lisbon").timezone

Result: "Europe/Lisbon"

Day time Duration

Data type duration(dt) has these attributes:

- days as number
- hours as number [0..23]
- minutes as number [0..59]
- seconds as number [0..59]

Getting the days of a day time duration

duration("P3DT2H30M").days

Result: 3

Getting the hours of a day time duration

duration("P3DT2H30M").hours

Result: 2

Getting the minutes of a day time duration

duration("P3DT2H30M").minutes

Result: 30

Getting the seconds of a day time duration

duration("P3DT2H30M").seconds

Result: 0

Year month Duration

Data type duration(ym) has these attributes:

- years as number
- months as number [0..11]

Getting the years of a year month duration

duration("P25M").years

Result: 2

Getting the months of a year month duration

```
duration("P4Y6M").months
```

Result: 6

Functions

Now

The now() function in the 🔗 DMN FEEL is used to retrieve the current date and time as a date and time data type. This function is essential for decision logic that needs to reference the present moment.

Basic Concepts

Representation: The now() function is represented in the format yyyy-MM-dd'T'HH:mm:ssZ, where the **Z** indicates the **timezone**.

Getting the Current Date and Time

```
now()
```

Result example: "2024-06-11T08:48:56.187237227Z"

Coordinated Universal Time (UTC)

The returned date and time indicates that the time is in **UTC**. You can use this information directly, or convert it to a **local timezone**.

Practical Applications

Adding 2 hours and 30 minutes to the current time:

```
now() + duration("PT2H30M")
```

Result: Current date and time plus 2 hours and 30 minutes

Calculating Time Since an Event

```
now() - date and time("2024-06-10T23:59:59Z")
```

Result: Current date and time minus a given date and time

Getting Specific Attributes of the Current Date and Time

```
{
   "year": now().year,
   "month": now().month,
   "day": now().day,
   "hour": now().hour,
   "minute": now().minute,
   "second": now().second,
   "weekday": now().weekday
}
```

Result: Individual components of the current date and time

Today

The today() function in the 🔗 DMN FEEL is used to retrieve the current date as a date data type (without the time component). This function is essential for decision logic that needs to reference just the current date.

Basic Concepts

Representation: The today() function is represented in the format yyyy-MM-dd.

Getting the Current Date

```
today()
```

Result example: "2024-06-11"

Practical Applications

Adding 1 year and 20 days to the Current Date:

```
today() + duration("P385D")
```

Result: Current date plus 365 + 20 days

Calculating Days Until a Specific Date

```
date("2035-06-10") - today()
```

Result: Current date and time minus a given date and time

Getting Specific Attributes of the Current Date

```
{
    "year": today().year,
    "month": today().month,
    "day": today().day,
    "weekday": today().weekday
}
```

Result: Individual components of the current date and time

Day, Week, Month Of

In this lesson, we will cover several important date functions in 🔗 FEEL used in Camunda, specifically:

- day of week()
- day of year()
- week of year()
- month of year()
- last day of month()

These functions help extract specific components from both date and date and time values.

Day of the Week

The day of week() function is used to retrieve the day of the week for a given date. Note that it always returns the English name of the day as a string.

Getting the Day of the Week

```
day of week(date("2025-12-31"))
```

Result example: "Wednesday"

Day of the Year

The day of year() function is used to retrieve the day of the year for a given date. The result is a number between **1 and 366**, accounting for leap years.

Getting the Day of the Year

```
day of year(date("2025-12-31"))
```

Result: 365

Week of the Year

The week of year() function is used to retrieve the week number of the year for a given date. The first week of the year is the one containing the **first Thursday of the year (ISO-8601)**.

Getting the Week of the Year

week of year(date("2025-12-31"))

Result: 1

Month of the Year

The month of year() function is used to retrieve the month of the year for a given date. Note that it always returns the English name of the month as a string.

Getting the Month of the Year

month of year(date("2025-12-31"))

Result: December

Last Day of Month

The last day of month() function is used to retrieve the last day of this month for a given date. The result is a date.

Getting the Last Day of the Month

last day of month(date("2025-12-01"))

Result: "2025-12-31"

Review

Through this course, you have learned how to evaluate **FEEL Temporal** data types, expressions and functions used in Camunda.

What Did I Learn?

You should now be able to:

- Understand the basic temporal data types:
    - date
    - time
    - date and time
    - day-time duration
    - year-month duration

- Operate with various temporal expressions:
    - **Addition**
    - **Subtraction**
    - **Multiplication**
    - **Division**
- Recognize most common temporal attributes:
    - year, years
    - month, months
    - day, days
    - weekday
    - hour, hours
    - minute, minutes
    - second, seconds
    - time offset
    - timezone
- Use various temporal functions:
    - now()
    - today()
    - day of week()
    - day of year()
    - week of year()
    - month of year()
    - last day of month()