



Advanced Query Language (DSL)

The Advanced Query feature uses a Domain-Specific Language (DSL) to create dynamic, personalized filters for articles and recommendations. This powerful system allows you to build complex logic using variables, primitives, functions, and conditional statements to deliver relevant content based on user data and context.

Key Variables

These built-in variables provide essential context for your queries:

- **EMPTY**: Represents an empty result set. Use when you want to return no results under certain conditions.

Example: `unread(articles) if user.age >= 18 else EMPTY`

Returns unread articles for users 18+, otherwise returns nothing.

- **NOW**: Current date and time timestamp.
- **TODAY**: Current date (without time component).
- **articles**: Collection of all available articles. Use this for article-specific queries like `unread(articles)`, `limit(articles, 5)`, or `filter(articles, tags="HR")`.
- **recommendations**: Collection of all available recommendations. Use this for recommendation-specific queries like `limit(recommendations, 3)`, or `filter(recommendations, tags="exercise")`.

Data Primitives

Access user health data through the `data` object structure:

Basic Metrics

- **data.<primitive>**: Access any recorded health metric (HeartRate, Weight, Temperature, etc.)

- `data.<primitive>.value`: The actual recorded value (number or decimal)
- `data.<primitive>.flags`: Severity level indicators with three levels:
 - `flags.gray`: Normal range
 - `flags.amber`: Warning level
 - `flags.red`: Critical level

Complex Metrics

For metrics with multiple values:

- `data.BloodPressure.systolicValue`
- `data.BloodPressure.diastolicValue`

Functions

Core Functions

- `limit(collection: array, count: int)`

Restricts the number of returned items from the specified collection.

Examples:

- `limit(unread(articles), 2)` - Shows maximum 2 unread articles
- `limit(recommendations, 3)` - Shows maximum 3 recommendations
- `unread(articles: array)`

Filters to show only unread articles.

Example:

```
unread(articles)
```

Returns all unread articles.

- `filter(collection: array, tags: string)`

Filters items by specific tags from the specified collection.

Examples:

- `filter(articles, tags="HR")` - Shows articles tagged with "HR"
- `filter(recommendations, tags="exercise")` - Shows recommendations tagged with "exercise"

- `random(collection: array, count: int)`

Returns a random selection from the specified collection.

Examples:

- `random(articles, 3)` - Shows 3 randomly selected articles
- `random(recommendations, 2)` - Shows 2 randomly selected recommendations

Conditional Logic

The DSL uses Python-style syntax for building complex conditions:

If/Else Statements

Syntax: `<result_if_true> if <condition> else <result_if_false>`

Example:

```
filter(articles, tags="HR") if data.HeartRate.flags.red > 0 else  
limit(unread(articles), 2)
```

Shows heart rate articles if the last reading is critical, otherwise shows 2 unread articles.

Logical Operators

- `and`: Both conditions must be true
- `or`: Either condition can be true

Example:

```
unread(articles) if data.HeartRate.flags.amber or  
data.HeartRate.flags.red else random(recommendations, 2)
```

Returns unread articles if heart rate is in warning or critical range, otherwise shows 2 random recommendations.

Combination Operator

- : Combines multiple article sets

Example:

```
filter(articles, tags="HR") + filter(recommendations, tags="sport")
```

Shows all articles tagged with "HR" plus all recommendations tagged with "sport".

Practical Examples

Basic Filtering

```
limit(unread(articles), 5)  
limit(recommendations, 3)
```

Show 5 most recent unread articles, or 3 most recent recommendations.

Health-Based Filtering

```
filter(articles, tags="diabetes") if data.BloodSugar.flags.red else  
unread(articles)
```

Show diabetes articles if blood sugar is critical, otherwise show all unread articles.

Complex Conditions

```
filter(articles, tags="exercise") if data.HeartRate.value > 100 and  
data.Weight.flags.amber else limit(articles, 3)
```

Show exercise articles if heart rate is elevated and weight is in warning range, otherwise show 3 articles.

Combining Collections

```
filter(articles, tags="nutrition") + filter(recommendations,  
tags="diet")
```

Show nutrition articles combined with diet recommendations.

This guide provides a comprehensive understanding of how to leverage the Advanced Query feature using DSL. Use these tools to create personalized, context-aware queries that can target either articles or recommendations based on user health data and preferences.



[Edit this page](#)