How to configure triggers in Xurrent

This article describes how to configure triggers in Xurrent to control which requests are synchronized.

Triggers in Xurrent: Triggers use field-based queries to filter requests. In this article we will see which fields, operators we can use in triggers.

Xurrent Request Filter Syntax Guide

Complete reference for filtering Xurrent requests in trigger conditions.

Filter Operators

All filters are applied as query parameters. Multiple conditions are joined using '**&**'.

Operator	Description	Example	Use Case
`=`	Equal (exact match)	`?id=43523`	Match exact value
`=!`	Not equal	`?status=!assigned`	Exclude specific value
`=` (empty)	Is null/empty	`?team_id=`	Field has no value
`=!` (empty)	Is not null	`?team_id=!`	Field has a value
`=value1,value2`	In (multiple values)	`?id=1,2,3,4`	Match any of the values
`=!value1,value2`	Not in	`?status=!assigned,accepted`	Exclude any of the values
`=<value`	Less than	`?created_at=<2016-01-15T23:00:00Z`	Value is less than
`=<=value`	Less than or equal	`?created_at=<=2016-01-15T23:00:00Z`	Value is less or equal
`=>value`	Greater than	`?completed_at=>2016-01-15T23:00:00Z`	Value is greater than
`=>=value`	Greater than or equal	`?updated_at=>=2016-01-15T23:00:00Z`	Value is greater or equal

Operator	Description	Example	Use Case
`=>value1<value2`	Range (exclusive)	`?created_at=>2015-12-31<2016-02-01`	Between two values
`=>=value1<=value2`	Range (inclusive)	`?created_at=>=2016-01-01<=2016-01-31`	Between inclusive bounds

Filterable Fields

Field	Type	Description	Operators
`id`	Integer	Request ID	`=` , `=!` , `in` , `not in` , `<` , `>` , etc.
`subject`	String	Request subject (max 255)	`=` , `=!` , `=` (empty), `=!` (empty)
`category`	Enum	Request category (required)	`=` , `=!` , `in` , `not in`
`impact`	Enum	Business impact	`=` , `=!` , `in` , `not in`
`status`	Enum	Request status	`=` , `=!` , `in` , `not in`
`service_instance`	Reference (ID)	Service instance	`=` , `=!` , `=` (empty), `=!` (empty)
`requested_by`	Reference (ID)	Person who requested	`=` , `=!` , `=` (empty), `=!` (empty)
`requested_for`	Reference (ID)	Person request is for	`=` , `=!` , `=` (empty), `=!` (empty)
`created_by`	Reference (ID)	Person who created	`=` , `=!` , `=` (empty), `=!` (empty)
`team`	Reference (ID)	Assigned team	`=` , `=!` , `=` (empty), `=!` (empty)
`member`	Reference (ID)	Assigned person	`=` , `=!` , `=` (empty), `=!` (empty)
`template`	Reference (ID)	Request template	`=` , `=!` , `=` (empty), `=!` (empty)
`organization`	Reference (ID)	Organization	`=` , `=!` , `=` (empty), `=!` (empty)
`created_at`	DateTime	Creation timestamp	`=` , `<` , `>` , `<=` , `>=` , ranges
`updated_at`	DateTime	Last update timestamp	`=` , `<` , `>` , `<=` , `>=` , ranges
`completed_at`	DateTime	Completion timestamp	`=` , `<` , `>` , `<=` , `>=` , ranges

Non-Filterable Fields

From the General Filtering Documentation: https://developer.xurrent.com/v1/general/filtering/

Following fields are not filterable:

1. Tags

2. Custom-Fields

Field-Specific Syntax

1. ID (Integer)

Format: `id=<value>` or `id=<value1>,<value2>`

```
# Single ID
id=43523

# Multiple IDs
id=1,2,3,4

# Exclude specific IDs
id=!100,200

# Greater than ID
id=>1000

# Range
id=>=100<=500
```

2. Status (Enum)

Format: `status=<value>`

Valid Values:

- `declined` , `on_backlog` , `assigned` , `accepted` , `in_progress` , `waiting_for` ,
- `waiting_for_customer` , `reservation_pending` , `workflow_pending` , `project_pending` , `completed`

```
# Single status
status=assigned

# Multiple statuses (OR)
status=assigned,accepted,in_progress

# Exclude statuses
status=!completed,declined

# Only open requests (not completed)
status=!completed
```

3. Category (Enum - Required)

Format: `category=<value>`

Valid Values:

- `incident` , `rfc` (Request for Change), `rfi` (Request for Information),
- `reservation` , `order` , `fulfillment` , `complaint` , `compliment` , `other`

```
# Single category
category=incident

# Multiple categories
category=incident,rfc,rfi

# Everything except incidents
category=!incident
```

4. Impact (Enum)

Format: `impact=<value>`

Valid Values: `low` , `medium` , `high` , `top`

```
# Single impact
impact=high

# High or top priority
impact=high,top

# Not low priority
impact=!low
```

5. Service Instance (Reference - ID)

Format: `service_instance=<id>`

Important: You must use the **numeric ID**, not the service instance name.

```
# Specific service instance
service_instance=12345

# Multiple service instances
service_instance=100,200,300

# Has a service instance (not null)
service_instance=!

# No service instance (null)
service_instance=

# Exclude specific service instance
service_instance=!12345
```

How to Get Service Instance ID:

1. Via API: `GET /v1/service_instances?name=YourServiceName`

2. From UI: Check the URL when viewing the service instance

3. From request details: Look at the `service_instance.id` field

---

6. Team (Reference - ID)

Format: `team=<id>`

```
# Specific team
team=42

# Multiple teams
team=10,20,30

# Has assigned team
team=!

# No assigned team
team=

# My team's requests
/requests/assigned_to_my_team
```

---

7. Member (Reference - ID)

Format: `member=<id>`

```
# Assigned to specific person
member=567

# Assigned to me
/requests/assigned_to_me

# Multiple members
member=100,200,300

# Has assignee
member=!

# Unassigned
member=
```

---

8. Requested By / Requested For (Reference - ID)

Format: `requested_by=<id>` or `requested_for=<id>`

```
# Requested by specific person
requested_by=789

# Requested for specific person
requested_for=456

# Requests I submitted
requested_by=<my_person_id>

# Requests on my behalf
requested_for=<my_person_id>

# Has requester
requested_by=!

# Missing requester (shouldn't happen as it's required)
requested_by=
```

9. Created At / Updated At / Completed At (DateTime)

Format: `YYYY-MM-DDTHH:MM:SSZ` (ISO 8601 in UTC)

```
# Exact datetime
created_at=2016-01-15T23:00:00Z

# After specific date
created_at=>2016-01-15T00:00:00Z

# Before specific date
created_at=<2016-12-31T23:59:59Z

# Date range (exclusive)
created_at=>2016-01-01T00:00:00Z<2016-02-01T00:00:00Z

# Date range (inclusive)
created_at=>=2016-01-01T00:00:00Z<=2016-01-31T23:59:59Z

# Last 24 hours
created_at=>2025-11-04T00:00:00Z

# Updated today
updated_at=>=2025-11-05T00:00:00Z

# Completed this month
completed_at=>=2025-11-01T00:00:00Z<=2025-11-30T23:59:59Z

# Has completion date
completed_at=!

# Not completed (no completion date)
completed_at=
```

10. Subject (String)

Format: `subject=<value>` (case-sensitive)

```
# Exact subject match
subject=Email%20server%20down

# Has a subject (not empty)
subject=!

# Empty subject (shouldn't happen as it's required)
subject=

# Note: Partial matching not supported - must be exact
```

11. Template (Reference - ID)

Format: `template=<id>`

```
# Created from specific template
template=100

# Created from templates
template=!

# Not from template (manual creation)
template=
```

12. Organization (Reference - ID)

Format: `organization=<id>`

```
# Specific organization
organization=50

# Has organization
organization=!

# No organization
organization=
```

Complete Examples

Example 1: High Priority Incidents Assigned to My Team

```
category=incident&impact=high,top&team=42&status=!completed
```

Example 2: Service-Specific Open Requests

```
service_instance=12345&status=assigned,accepted,in_progress&category=incident,rfc
```

Example 3: Recently Created Unassigned Requests

```
created_at=>2025-11-01T00:00:00Z&member=&status=!completed
```

Example 4: Completed Requests This Month

```
status=completed&completed_at=>=2025-11-01T00:00:00Z<=2025-11-30T23:59:59Z
```

Example 8: Requests Created in Last 7 Days for Specific User

```
created_at=>2025-10-29T00:00:00Z&requested_for=789&status=!declined
```

Example 9: High Impact RFCs Pending Approval

```
category=rfc&impact=high,top&status=assigned,accepted
```

Best Practices

1. Always Use IDs for References

- **DON'T:** `service_instance=Production Database`
- **DO:** `service_instance=12345`

To find IDs, use the API:

```
# Get service instance ID
GET /v1/service_instances?name=Production%20Database

# Get team ID
GET /v1/teams?name=IT%20Support

# Get person ID
GET /v1/people?primary_email=user@example.com
```

2. Combine Multiple Conditions

All conditions are joined with AND:

```
category=incident&impact=high&status=assigned&team=42
# Means: incident AND high impact AND assigned AND team 42
```

3. DateTime Filtering Tips

- Always use UTC timezone (`Z` suffix)
- Use ISO 8601 format: `YYYY-MM-DDTHH:MM:SSZ`
- For "today", calculate the start of day: `2025-11-05T00:00:00Z`
- For ranges, use inclusive bounds: `>=start<=end`

4. Handle Null Values

- Check if field has value: `field=!`
- Check if field is empty: `field=`

- Example: Find unassigned: `member=`

Quick Reference Table

Need	Filter Syntax	Example
Specific request	`id=<number>`	`id=12345`
Multiple requests	`id=<n1>,<n2>`	`id=100,200,300`
By service	`service_instance=<id>`	`service_instance=12345`
By status	`status=<value>`	`status=assigned`
Not completed	`status=!completed`	`status=!completed`
High priority	`impact=high,top`	`impact=high,top`
My team	`team=<id>`	`team=42`
Unassigned	`member=`	`member=`
Has assignee	`member=!`	`member=!`
Created today	`created_at=><date>`	`created_at=>2025-11-05T00:00:00Z`
Incidents only	`category=incident`	`category=incident`
Major incidents	`major_incident_status=accepted`	`major_incident_status=accepted`

More information

- How to Sync Requests in Xurrent

- Set Up a Connection in Basic Mode for Xurrent

Have more questions? Ask the community