Reglas de filtrado

Los permisos, la validación y el parámetro filter de la API se basan en una estructura JSON específica para definir sus reglas. En esta página se describe la sintaxis para crear reglas de filtro planas, relacionales o complejas.

Sintaxis

- Campo: cualquier campo raíz, campo relacional u operador lógico válido
- Operador: cualquier operador de filtro válido
- Valor: cualquier valor estático válido o variable dinámica

```
{
  <field>: {
    <operator>: <value>
  }
}
```

Ejemplos

```
{
  "title": {
    "_contains": "Catalogo"
  }
}
```

```
{
  "owner": {
    "_eq": "$CURRENT_USER"
  }
}
```

```
{
  "datetime": {
    "_lte": "$NOW"
  }
}
```

```
{
  "category": {
```

```
□□"_null": true
□}
}
```

Operadores de filtro

Nombre del operador (en la aplicación)	Operador	Descripción
Igual a	`_eq`	Igual a
No es igual a	`_neq`	No es igual a
Menor que	`_lt`	Menor que
Menor o igual que	`_lte`	Menor o igual que
Mayor que	`_gt`	Mayor que
Mayor o igual que	`_gte`	Mayor o igual que
Es uno de los	`_in`	Coincide con cualquiera de los valores
No es uno de los	`_nin`	No coincide con ninguno de los valores
Es nulo	`_null`	Es nulo `null`
No es nulo	`_nnull`	No es nulo `null`
Contiene	`_contains`	Contiene la subcadena
Contiene (no distingue entre mayúsculas y minúsculas)	`_icontains`	Contiene la subcadena que no distingue entre mayúsculas y minúsculas
No contiene	`_ncontains`	No contiene la subcadena
Comienza con	`_starts_with`	Comienza con
Comienza con (sin distinción entre mayúsculas y minúsculas)	`_istarts_with`	Comienza con, sin distinción entre mayúsculas y minúsculas
No comienza con	`_nstarts_with`	No comienza con
No comienza con (sin distinción entre mayúsculas y minúsculas)	`_nistants with`	No comienza con, sin distinción entre mayúsculas y minúsculas
Termina con	`_ends_with`	Termina con
Termina con (sin distinción entre mayúsculas y minúsculas)	`_iends_with`	Termina con, sin distinción entre mayúsculas y minúsculas
No termina con	`_nends_with`	No termina con
No termina con (sin distinción entre mayúsculas y minúsculas)	`_niends_with`	No termina con, sin distinción entre mayúsculas y minúsculas
Está entre	`_between`	El valor interseca un punto dado

Nombre del operador (en la aplicación)	Operador	Descripción
No está entre	`_nbetween`	El valor no interseca un punto dado
Está vacío	`_empty`	Está vacío (`null` o falso)
No está vacío	`_nempty`	No está vacío (`null` o falso)

Relacional

Puede apuntar a valores relacionados anidando nombres de campo. Por ejemplo, si tiene un campo relacional Many-to-One, puede establecer una regla para el campo mediante la siguiente sintaxis.authorauthor.name

```
{
□"author": {
□□"name": {
□□□"_eq": "Rijk van Zanten"
□□}
□}
}
```

Cuando se utilizan relaciones M2M, se creará una tabla de uniones y el filtro se aplicará a la propia tabla de uniones. Por ejemplo, si tiene una colección, con una relación M2M con los autores de cada libro, es probable que la colección de cruces tenga un nombre y 3 campos: y . Para filtrar libros específicos en función de sus autores, debe pasar por la tabla de uniones y el campo :books books_authorsid books_id authors_id authors_id

```
{
□"authors": {
□□"authors_id": {
□□□"name": {
□□□□"_eq": "Rijk van Zanten"
□□□}
□□}
□}
}
```

Operadores lógicos

Puede anidar o agrupar varias reglas mediante los operadores lógicos o. Cada operador lógico contiene una matriz de reglas de filtro, lo que permite un filtrado más complejo. Tenga en cuenta también en el ejemplo que los operadores lógicos se pueden subanidar en operadores lógicos. Sin embargo, no se pueden subanidar en reglas de filtro._and_or

```
{
  "_or": [
    {
      "_and": [
        {
          "user_created": {
            "_eq": "$CURRENT_USER"
          }
        },
        {
          "status": {
            "_in": ["published", "draft"]
          }
        }
      ]
    },
    {
      "_and": [
        {
          "user_created": {
            "_neq": "$CURRENT_USER"
          }
        },
        {
          "status": {
            "_in": ["published"]
          }
        }
      ]
    }
  ]
}
```

Algunos vs Ninguno en uno a muchos

Al aplicar filtros a un campo de uno a varios, la plataforma usará de forma predeterminada una búsqueda de "algunos", por ejemplo, en:

```
{
  "categories": {
    "name": {
      "_eq": "Recipe"
    }
  }
}
```

El elemento primario de nivel superior se devolverá si una de las categorías tiene el nombre . Este comportamiento se puede invalidar mediante el uso de los operadores y explícitos, por ejemplo: Recipe_some_none

```
{
  "categories": {
    "_none": {
      "name": {
        "_eq": "Recipe"
      }
    }
  }
}
```

obtendrá todos los elementos principales que no tengan la categoría "Receta".

Revisión #1
Creado 2025-07-23 02:24:09 UTC
Actualizado 2025-07-23 02:34:28 UTC