

6. GraphQL

I tillegg til REST-APIet tilbyr FEED et GraphQL-endepunkt, primært beregnet på produktsøk med avansert filtrering. Autentisering er identisk med REST – Bearer-token i `Authorization`-headeren. Krevder `export-authority`.

Endepunkt	Beskrivelse
<code>/graphql</code>	GraphQL-endepunkt
<code>/graphiql</code>	Interaktivt GraphQL-grensesnitt (browser)

GraphiQL er nettlesebasert og lar deg skrive og kjøre spørringer manuelt – nyttig for utforskning og feilsøking.

Tilgjengelige queries

Det finnes én query: `products`. Den tar følgende argumenter:

Argument	Type	Standard	Beskrivelse
<code>productSearch</code>	<code>ProductSearchInput</code>	–	Filtreringskriterier
<code>sort</code>	<code>[ProductSortInput!]</code>	–	Sortering
<code>page</code>	<code>Int</code>	0	Sidenummer (0-basert)
<code>size</code>	<code>Int</code>	20	Antall produkter per side

Responsen (`ProductPage`) inneholder `content`, `pageNumber`, `pageSize`, `totalElements` og `totalPages`.

Sortering

Sortering angis som en liste med objekter: `{ property: "productNo", direction: ASC }`. Støtter `languageCode` for flerspråklig sortering.

Filtrering

GraphQL-søket støtter rik filtrering via `ProductSearchInput`:

Produktthode

- `productNo`, `altProductNo`, `name`, `altName`
- `productType` – `BASE`, `PRODUCT` eller `VARIANT`
- `productStatusIds`
- `modifiedTimeFrom` – kun produkter endret etter gitt tidspunkt

Attributter – via `AttributeFilterInput`

Operator	Beskrivelse
<code>EQ</code>	Lik en verdi
<code>IN</code>	Én av flere verdier
<code>GT/GTE</code>	Større enn / større enn eller lik
<code>LT/LTE</code>	Mindre enn / mindre enn eller lik
<code>BETWEEN</code>	Innenfor et verdiintervall
<code>EXISTS</code>	Attributtet har en verdi
<code>MISSING</code>	Attributtet mangler verdi

Enheter – filtrer på entitetstype og/eller spesifikke enheter, eller finn produkter som mangler enheter.

ETIM – filtrer på versjon, klasse-ID og feature-verdier (`EQ`, `HAS_VALUE`, `MISSING_VALUE`).

Media – filtrer på manglende bilde, dokument eller medieroller (med `AND/OR`-logikk på roller).

Relasjoner – filtrer på relasjonstype, eller finn produkter som mangler en gitt relasjon.

Feltseleksjon

En fordel med GraphQL er at du kun henter feltene du faktisk trenger. For flerspråklige felt angir du hvilke språk du vil ha, f.eks. `name(languageCodes: ["nb", "en"])`. Det samme gjelder `attributes`, `entityTypes`, `etim`, `relations` og `packaging` – alle støtter ID-basert filtrering direkte i feltet.

Eksempelspørring

```
query {
  products(
    productSearch: {
      productType: PRODUCT
      modifiedTimeFrom: "2025-01-01T00:00:00Z"
    }
    sort: [{ property: "productNo", direction: ASC }]
    page: 0
    size: 20
  ) {
    content {
      id
      productNo
      name(languageCodes: ["nb"]) {
        languageCode
        value
      }
    }
    totalElements
    totalPages
  }
}
```

Feilhåndtering

GraphQL returnerer alltid HTTP 200, men feil legges i `errors`-feltet i responsen:

```
{
  "data": null,
  "errors": [
    {
      "message": "Invalid argument",
      "extensions": { "classification": "BAD_REQUEST" }
    }
  ]
}
```