

# Dokumentasjon API

---

**DIREKTORATET FOR HØYERE UTDANNING OG  
KOMPETANSE**

HK-dir  
03.12.2025

## Innhold

Formål og målgruppe.....	2
Kort om API-løsningen .....	2
Tilgang og autentisering .....	3
Bruksvilkår .....	3
Tolking av data .....	4
Endring av feltnavn og nye felter .....	4
Veier inn til data .....	4
Mindre datasett: .....	4
Utlisting av hele tabell:.....	4
API konsepter og spørringsspråk.....	5
Praktiske råd (kvalitet, ytelse, drift) .....	5
GDPR og skjerming.....	6
Eksempel på Arbeidsflyt (fra idé til rapport) .....	6
Vanlige feilsituasjoner.....	6
Query .....	6
GroupBy.....	7
Filter .....	7
Eksempler på queries .....	8
Eksempel på query med forklaring .....	8
Eksempel på bruk av «exclude» i filter .....	8
Eksempel på bruk av «top» i filter.....	9
Eksempel på bruk av «all» i filter .....	9
Eksempel på bruk av «item» i filter .....	10
Eksempel på bruk av «between» i filter .....	10
Eksempel på bruk av «like» i filter .....	11
Eksempel på bruk av «lessthan» i filter .....	11
Eksempel på enkel egendefinert query .....	12
Eksempel på query med mange filtre .....	13
Eksempler på uthenting av data.....	14
Programmeringsspråket R: Hente JSON og skriv til CSV.....	14
R: Robust streaming av CSV.....	15

Python: Requests + pandas .....	16
Programmeringsspråket Python: Direkte til JSON.....	17
Eksempel på bruk via AI-assistent .....	17
Prompt (naturlig språk) mot en tabell du har hentet fra DBH-API .....	18
Forslag til JSON spørring .....	18
cURL (valgfridd) basert på samme JSON .....	18
Hente kanalregisterdata. ....	19
Hente alle tidsskrift som er registrert eller endret for en bestemt dag: .....	19
For å hente alle tidsskrift med endringer mellom to datoer: .....	19
Hente to tidsskrift(Du kan også søke på et. Tidsskrift er kommaseparert):.....	19
Hente alle tidsskrift: .....	19
Stier: .....	20
BULK .....	20
CURL script for å hente tidsskrift:.....	20

## Formål og målgruppe

Dette dokumentet gir en samlet, lettlest og samtidig teknisk innføring i API-løsningen for DBH og DBH-HYU.

Målgruppen er både utviklere (for integrasjoner, automatisering og datavarehus) og fagpersoner uten utviklerbakgrunn som ønsker å hente ut statistikk fra DBH på en trygg og effektiv måte.

## Kort om API-løsningen

Dette APIet er et programmerbart grensesnitt levert av Direktoratet for høyere utdanning og kompetanse (HK-dir) for å hente data om høyere utdanning.

APIet tilbyr statistikk om bl.a. organisasjoner, studietilbud, studenter, doktorgrader, økonomi, publiseringskanaler, personaldata og areal, både fra universiteter/høyskoler og fagskoler.

Uttrekket som vises i bildet, er basert på filtrering av data. Hvert felt er definert i egne seksjoner med tilhørende syntaks. item, all, between, greaterthan, top m.fl. Se utvalget filtervariabler i eget avsnitt.

Det finnes to typer tabeller: listetabeller og matematiske tabeller.

I 001-innhold identifiseres matematiske tabeller ved at "Math" = "1". Disse **må** inneholde feltet "groupBy". I groupBy kan du angi så mange felt du ønsker, men det er ikke tillatt å bruke \* eller [] (uten innhold).

Listetabeller må inneholde : "variabler":

Groupby og variabler seksjon kan aldri være [] eller ikke eksistere.

I listetabeller kan du derimot bruke \*, for eksempel slik: "variabler": ["\*"].

I groupBy og variabler legger du inn de feltene du ønsker skal vises i tabellen.

Statuslinje kan settes til "N" dersom leveringsinformasjon ikke skal tas med over i Excel.

Alle spørringer må defineres enten som listetabeller eller matematiske tabeller. Dette er nærmere beskrevet nedenfor.

Data leveres over HTTPS med UTF-8-koding, i enten JSON- eller CSV-format.

## Tilgang og autentisering

- Tilgang: Åpne data kan hentes uten innlogging. Persondata og små tall (mindre enn 3) skjermes etter GDPR. Alle endepunkt ligger tilgjengelig i API klient.
- Du kan få tilgang til nøyaktige tall for din egen institusjon med en JSON Web Token (JWT). For å få en slik token må du være registrert i Feide. Du kan kontakte oss på [dbh@hkdir.no](mailto:dbh@hkdir.no) for mer veiledning om dette

## Bruksvilkår

Vilkårene for bruk følger til enhver tid gjeldende Norsk lisens for offentlige data. Legg spesielt merke til følgende: HK-dir/DBH/DBH-HYU tar ikke ansvar for konsekvenser som følge av feil i systemet.

DBH/DBH-HYU skal oppgis som kilde ved bruk av data fra tjenesten. Dataene skal ikke presenteres på en måte som fremstår som villedende, og heller ikke fordreies eller fremstilles uriktig.

Les grundig alle vilkårene for bruk i den standardiserte [lisensen for offentlige data](#). Alle forespørsler logges av HK-dir av hensyn til bruksmønster og drift.

## Tolking av data

Egenskaper ved data benyttes til å aggregere og kategorisere data, som fordeling på finansieringskategorier, studienivå, heltid/deltid etc. Datakvaliteten er derfor avgjørende.

Endringer og overgangsperioder for data finnes i dokumentasjonen for hver enkelt tabell og vi viser ellers til dokumentasjon for hver enkelt tabell og ellers til DBH/DBH-HYUs tabelldokumentasjon når det gjelder tolking av data. Ved bruk av KI-verktøy som Chat GPT er det viktig å kontrollere analysen og sørge for at data blir framstilt riktig.

Eventuelle genereringer via KI må merkes som KI generert.

## Endring av feltnavn og nye felter

Vi vedlikeholder felt. Dokumentasjon på vedlikehold ligger under driftsmeldinger hvis du besøker klienten, alternativt kan du finne de i tabell 916 - API-driftsmeldinger.

Endringer kan innføres uten forvarsel selv om vi gir beskjed via e-post når det er store endringer. Innføring av nye tabeller ligger også i driftsmeldinger.

## Veier inn til data

- DBH API-klient: <https://dbh.hkdir.no/dbhapiklient> – bygg spørringer, test og last ned resultat.
- DBH-HYU API-klient: Kommer snart!
- Programmatisk tilgang: cURL/HTTP-klienter, R, JavaScript, Python m.m.
- Stier:  
**DBH data:** <https://dbh-data.dataporten-api.no/Tabeller/>  
**Kanalregister data (850..):** <https://kanalregister.hkdir.no/api/krtabeller/>  
**Mindre datasett:** /hentCSVtabellData eller /hentJSONtabellData  
**Større datasett:** /streamCsvData eller /streamJsonData  
**Utlisting av hele tabell:** /bulk-csv?rptNr=???

## API konsepter og spørringsspråk

Det finnes to typer tabeller. Rene utlistinger og tabeller som kan summeres via spørringer. Disse siste er merket med Math = 1 i innholdsfortegnelsen. Du kan selv bestemme hvor mange felt du vil ha med i utlistingen.

- tabell\_id: Hvilken tabell det hentes data fra (se 001-innhold/002-metadata/913–916).
- groupBy: Bestemmer aggregeringsnivå (som i SQL GROUP BY). For tabeller som du kan lage egne beregninger. Alle felt må finnes i groupby seksjonen i spørringen.
- variabler: Hvilke felter som leveres (særlig for «lister» uten groupBy). «\*» betyr alle.
- sortBy: Sorteringsrekkefølge; må være delmengde av groupBy når groupBy brukes.
- filter: Utvalg av data. Støtter bl.a. all, item, top, between, like, greaterthan, lessthan og exclude(ikke obligatorisk).
- statuslinje/kodetekst/begrensning: Styrer meta-informasjon, kodetekster og antall rader som returneres i klienten.

## Praktiske råd (kvalitet, ytelse, drift)

- Start alltid i API-klienten for å prøve ut spørringen – kopier så JSON til skript/kode.
- Bruk «item», «top», «between» m.fl. for å begrense uttrekk; nettlesere kan gå tom for minne.
- For store datamengder: foretrekk CSV for mindre filstørrelse, eller bruk stream-endepunktene til fil.
- Følg med på driftsmeldinger via tabell 916 (API-Driftsmeldinger) eller se eget felt på nettsiden for klienten.
- Kodetekster: Sett «kodetekst":"J» for å få forklarende navn på koder (øker responstid noe).
- Sorteringsregler: Felter i sortBy må ligge i groupBy (som i SQL).
- Datoformater: Følg eksemplene i dokumentasjonen; feil datoformat gir tomt treff eller feil.
- Kontroller tall: Sammenlign mot DBH-rapporter ved mistanke om dubletter/overaggregering.
- API klient: I noen tilfeller kan du hente mer data enn skjermbuffer tåler. Da vil det komme opp en blank side. Da må du dele opp spørringen. Eventuelt prøve funksjonen: 'Direkte Csv'.

## GDPR og skjerming

For tabeller merket for skjerming (001-Innhold) anonymiseres små tall (<3) for brukere uten institusjonstilgang. Når data presenteres videre, vil det fremgå at tall kan være skjermet, for å unngå feiltolkning. Institusjoner må logge på Feide for å få nøkkel(token) slik at de kan se små tall der disse tallene er skjermet. Institusjonene henter token fra Feide som de sender med spørringen.

DBH api klient anonymiserer data. Her kan ikke bruker logge på Feide. Brukere må lage script for å hente små tall (som beskrevet over).

## Eksempel på Arbeidsflyt (fra idé til rapport)

1. Identifiser tabell(er) og variabler (se 001-innhold/002-metadata/913–916).
2. Bygg og test spørring i API-klienten; legg inn nødvendige filter.
3. Kjør mot /hentJSONTabellData eller /hentCSVTabellData– verifiser datakvalitet.
4. For store uttrekk/produksjon: bruk / /streamCsvData og lagre til fil. Du kan også bruke BULK og kjøre ut en hel tabell. BULK gjelder for tabeller som ikke er beskyttet av GDPR. Disse er merket med Bulk tabell i tabell 001-innhold.
5. Import til Excel: Bruk «Data → Fra tekst», velg UTF-8 og semikolon som skilletegn.

## Vanlige feilsituasjoner

- Ingen/feil data: Sjekk filterverdier, datoformat, at sortBy  $\subseteq$  groupBy, og at «tabell\_id» er korrekt mot forslaget som du finner i api klienten.
- Små avvik mot rapporter: Kan skyldes avrunding i rapporter eller ulike filtre/valg. Fjern avrunding i rapport ved behov.
- 1M-grense i klient: Del opp spørringen eller bruk stream-endepunkter.
- Studieprogramnavn vises ikke: Krever årstall, semester og studieprogramkode i spørringen.
- Nettleser-minne: Begrens utvalg (item/top/between) eller bruk stream til fil.

## Query

Dette er forespørselen som skal gi et resultat tilbake. Den er laget i JSON format og inneholder flere sammensatte grupperinger som påvirker spørringen og aggregering av

data. Det er et lite språk med en egen syntaks som er forklart i dette dokumentet. Det kommer forslag til bruk opp i klienten. For eksempel vil delen som inneholder gruppering bestemme nivået som vi kombinerer data på. Bruker kan føye til felt etter ønske ut fra variabel eller groupby listen.

Seksjonen – ‘groupby’ vil være summeringsnivået( slik som i SQL). Denne seksjonen vil også angi hvilke felt som kommer ut i rapporten utenom summeringsfelt.

For rapporter uten sum felt, kan vi lage tabellen mindre ved bruk av «variabler» seksjon og spesifisere hvilke variabler som skal returneres. Filterdelen av spørringen vil bestemme utvalget av data. Her vises også et utvalg og det mulig å føye til eller ta bort etter ønske.

Formelfelt (eksempel: Antall totalt) kan ikke legges i filter, men de fleste andre felt i variabellisten kan velges slik at bruker ikke trenger å hente alle feltene i tabellen. Sortering(sortby) fylles ut etter ønske. Alle fagdata blir returnert som alfanumeriske felt i JSON/CSV format.

## GroupBy

Eksempel groupby: ["Institusjonskode", "Avdelingskode", "Årstall"]. Det er samme regler som i SQL, siden koden blir konvertert til SQL. Bruker kan flytte inn og ut felt etter ønske. For eksempel vil noen ønske "Årstall" først. Da flytter de det feltet først. Gruppering bestemmer også sorteringsorden på samme måte som i "Group By" i SQL.

Rapporter som inneholder summer, må inneholde minst ett felt i gruppering. Gruppering brukes til å lage ‘group by’ spørring mot databasen. Vi kan sjonglere med variablene i forhold til data. Etter gruppering kommer et avsnitt med forslag til utvalg av data. Vi kaller dette "Filter". Sammen med gruppering er dette en måte å konstruere data på. Vi kan ta bort eller legge til filter.

Goupby tabeller er merket med ‘Math’ = 1 i tabell 001-innhold.

## Filter

Eksemplene under er hentet fra skjermbildet i klienten. Det er et eget spørrespråk for DBH API. Foreløpig er der følgende filtervarianter: top, all, item, between, greaterthan og lessthan. Disse kan kombineres med ulike verdier. Disse er i anførselstegn("<nnnn>"), adskilt med komma.

Det er mulig å legge inn exclude i alle disse, unntatt ‘item’. For ‘item’, kommer heller ikke ‘exclude’ med som forslag. Exclude har form av en tabell. Den kan ha formen "exclude":

["xxxx"] eller "exclude": ["xxxx","xxxx","xxxx"]. Default er at den er tom, "exclude": [""].  
"Exclude» er ikke obligatorisk i querien.

Det må finnes minst et filter pr. tabell. Det er bare tabell\_id og api\_versjon som er obligatorisk i header.

## Eksempler på queries

### Eksempel på query med forklaring

```
{"tabell_id":133,"api_versjon":1,"statuslinje":"N","kodetekst":"J","desimal_separator":",", ←  
bare de to første feltene er obligatoriske.
```

```
"variabler":["Studentkategori"], ← '*' gir alle eller navngi felt slik som her
```

```
"sortBy":["Studentkategori"],
```

```
"filter":{
```

```
  "variabel": "Studentkategori",
```

```
  "selection": {
```

```
    "filter": "all", → se ellers 'infotag' i klient. Ofte item eller all
```

```
    "values": [
```

```
      "*" ← Gir alle. * brukes ved filter all. Ved filter "item" spesifiseres
```

```
verdier adskilt med komma.
```

```
    ]
```

```
  }
```

```
}]
```

```
}
```

### Eksempel på bruk av «exclude» i filter

```
"filter":{
```

```
  "variabel": "Tekst-egendefinert",
```

```
  "selection": {
```

```

    "filter": "all",
    "values": ["*"],
    "exclude": [
        "null" ← Her hentes alle unntatt de verdier som har null.
    ]
}
}
]

```

## Eksempel på bruk av «top» i filter

Her kan vi angi et tall og systemet vil hente de høyeste tre årstall. Verdier er alltid bare ett tall. Ikke flere. Dette valget egner seg for rullerende spørringer.

```

"filter":{
    "variabel": "Årstall",
    "selection": {
        "filter": "top",
        "values": ["3"]
    }
}
]

```

## Eksempel på bruk av «all» i filter

Her henter vi alle forekomster som er større enn null i database. Verdien i verdier er alltid "\*".

```

"filter":{
    "variabel": " Institusjonskode ",
    "selection": {
        "filter": "all",

```

```
        "values": ["*"]
      }
    }
  ]
```

## Eksempel på bruk av «item» i filter

Her henter vi alle forekomster som har verdier i listen slik som angitt i søket. Items er adskilt med komma.

```
"filter":{
  "variabel": "avdkode",
  "selection": {
    "filter": "item",
    "values": ["220760", "260250", "000000", "460000"]
  }
}
```

Vi kan også ha bare en verdi i rekken.

```
"filter":{
  "variabel": "semester",
  "selection": {
    "filter": "item",
    "values": ["3"]
  }
}
```

## Eksempel på bruk av «between» i filter

Typisk intervall mellom årstall. Tilsvarende 'between' i sql.

```
"filter":[{
  "variabel": "Årstall",
  "selection": {
    "filter": "between",
    "values": ["2010","2017"]
  }
}]
```

## Eksempel på bruk av «like» i filter

generisk søk. Tilsvarende 'like' i sql. % kan være foran, bak eller begge steder. Vi kan også bruke exclude i filteret.

```
"filter":[{
  "variabel": "Original tittel",
  "selection": {
    "filter": "like",
    "values": ["%Remote%"]
  }
}]
```

## Eksempel på bruk av «lessthan» i filter

Typisk årstall eldre enn et gitt år, men kan også brukes på andre variabler. Dette eksempelet henter alle institusjoner som har institusjonskode mindre enn "0500", unntatt institusjonskode "0264".

```
"filter":[{
  "variabel": "Institusjonskode",
  "selection": {
    "filter": "lessthan",
```

```
        "values": ["0500"],
        "exclude": ["0264"]
    }
}
]
```

## Eksempel på enkel egendefinert query

```
{"tabell_id":123,"api_versjon":1,"statuslinje":"N","kodetekst":"J","decimal_separator":",",
"groupBy":["Årstall","Institusjonskode", "Avdelingskode"],
"sortBy":["Årstall","Institusjonskode", "Avdelingskode"],
"filter":[{"
    "variabel": "Institusjonskode",
    "selection": {
        "filter": "item",
        "values": ["1150"],
        "exclude": [""]
    }
},{
    "variabel": "Årstall",
    "selection": {
        "filter": "top",
        "values": ["7"],
        "exclude": ["2013","2016"]
    }
}
}
]
```

## Eksempel på query med mange filtre

```
{
  "tabell_id": 60,
  "api_versjon": 1,
  "statuslinje": "N",
  "decimal_separator": "",
  "groupBy": ["Institusjonskode", "Avdelingskode", "Årstall", "Semester", "Alder"],
  "sortBy": ["Institusjonskode", "Avdelingskode", "Årstall"],
  "filter": [
    {
      "variabel": "Institusjonskode",
      "selection": {
        "filter": "all",
        "values": ["*"]
      }
    },
    {
      "variabel": "Avdelingskode",
      "selection": {
        "filter": "all",
        "values": ["*"]
      }
    },
    {
      "variabel": "Årstall",
      "selection": {
        "filter": "item",
        "values": ["2017"]
      }
    },
    {
      "variabel": "Semester",
      "selection": {
        "filter": "item",
        "values": ["3"]
      }
    }
  ]
}
```

```
    },  
    {  
      "variabel": "Alder",  
      "selection": {  
        "filter": "between",  
        "values": ["30","40"]  
      }  
    }  
  ]
```

I dette tilfellet kan du også dele opp query slik:

```
"filter":[{  
  "variabel": "Institusjonskode",  
  "selection": {  
    "filter": "between",  
    "values": ["0500","2000"]  
  }  
}]
```

## Eksempler på uthenting av data

### Programmeringsspråket R: Hente JSON og skriv til CSV

Eksempel på bruk av https og for å gjøre et POST-kall mot /hentJSONTabellData. Bytt ut tabell og filter etter behov. For tabeller med store datamengder kan /streamCsvData eller /bulk være mer robust.

```
# Pakker  
library(httr2)  
library(jsonlite)  
library(tibble)  
library(readr)
```

```

library(dplyr)

# Endepunkt
url <- "https://dbh.hkdir.no/api/hentJSONTabellData"

# (Valgfritt) token for institusjonstilgang
token <- Sys.getenv("DBH_JWT") # sett miljøvariabel hvis nødvendig

# Forespørsel (tilpass tabell_id, groupBy, sortBy og filter)
payload <- list(
  tabell_id = 60,
  api_versjon = 1,
  groupBy = c("Institusjonskode", "Årstall"),
  sortBy = c("Institusjonskode", "Årstall"),
  filter = list(
    list(variabel="Institusjonskode", selection=list(filter="item",
values=list("NTNU", "UiO"))),
    list(variabel="Årstall", selection=list(filter="between",
values=list("2021", "2024")))
  ),
  kodetekst = "J"
)

req <- request(url) |>
  req_headers("Content-Type" = "application/json") |>
  req_body_json(payload)

if (nzchar(token)) {
  req <- req |> req_headers(Authorization = paste("Bearer", token))
}

resp <- req_perform(req)
stop_for_status(resp)

# Tolke respons (tilpass etter responsformatet fra DBH for valgt tabell)
res_list <- resp_body_json(resp, simplifyVector = TRUE)

# Anta at data ligger i res_list$data (tilpass hvis annerledes)
df <- as_tibble(res_list$data)

# Skriv til CSV (semikolon brukes ofte i NO/Excel)
write_delim(df, "dbh_uttrekk.csv", delim=";", na = "", append = FALSE)

```

## R: Robust streaming av CSV

For store uttrekk: bruk `/streamCsvData` og les filen inn igjen i R.

```

library(httr2)
library(readr)

url <- "https://dbh.hkdir.no/api/streamCsvData"
payload <- list(

```

```

tabell_id    = 60,
api_versjon = 1,
groupBy      = c("Institusjonskode", "Årstall"),
sortBy       = c("Institusjonskode", "Årstall"),
filter       = list(
  list(variabel="Årstall", selection=list(filter="between",
values=list("2019", "2024")))
)
)

tmp <- tempfile(fileext = ".csv")

req <- request(url) |>
  req_headers("Content-Type" = "application/json") |>
  req_body_json(payload)

resp <- req_perform(req, path = tmp)
stop_for_status(resp)

df <- read_delim(tmp, delim=";", show_col_types = FALSE)

```

## Python: Requests + pandas

Eksempel på robust nedlasting via /streamCsvData. Tilpass «payload» og legg inn Authorization-header hvis du har institusjonstoken.

```

import os
import requests
import pandas as pd

URL = "https://dbh.hkdir.no/api/streamCsvData"
TOKEN = os.getenv("DBH_JWT") # sett miljøvariabel ved behov

payload = {
  "tabell_id": 60,
  "api_versjon": 1,
  "groupBy": ["Institusjonskode", "Årstall"],
  "sortBy": ["Institusjonskode", "Årstall"],
  "filter": [
    {"variabel": "Årstall", "selection": {"filter": "between",
"values": ["2020", "2024"]}}
  ],
  "kodetekst": "J"
}

headers = {"Content-Type": "application/json"}
if TOKEN:
  headers["Authorization"] = f"Bearer {TOKEN}"

out_path = "dbh_uttrekk.csv"
with requests.post(URL, json=payload, headers=headers, stream=True,

```

```

timeout=120) as r:
    r.raise_for_status()
    with open(out_path, "wb") as f:
        for chunk in r.iter_content(chunk_size=8192):
            if chunk:
                f.write(chunk)

# Les inn i pandas (semikolon-separert)
df = pd.read_csv(out_path, sep=";", encoding="utf-8")
print(df.head())

```

## Programmeringsspråket Python: Direkte til JSON

Hvis du bruker /hentJSONTabellData:

```

import requests
import pandas as pd

URL = "https://dbh.hkdir.no/api/hentJSONTabellData"
payload = {
    "tabell_id": 60,
    "api_versjon": 1,
    "groupBy": ["Institusjonskode", "Årstall"],
    "sortBy": ["Institusjonskode", "Årstall"],
    "filter": [
        {"variabel": "Institusjonskode", "selection": {"filter": "item",
"values": ["NTNU","UiO"]}},
        {"variabel": "Årstall", "selection": {"filter": "item", "values":
["2024"]}}
    ],
    "kodetekst": "J"
}

r = requests.post(URL, json=payload, timeout=60)
r.raise_for_status()
res = r.json()

# Anta at relevante rader ligger i res["data"] (tilpass til faktisk
struktur)
df = pd.json_normalize(res.get("data", []))
print(df.shape)

```

## Eksempel på bruk via AI-assistent

Du kan bruke en AI-assistent til å formulere en DBH-forespørsel i naturlig språk mot en tabell som du har hentet fra DBH api.

## Prompt (naturlig språk) mot en tabell du har hentet fra DBH-API

Lag en DBH-API-spørring som henter antall studenter per institusjon. F.eks tabell 60 for år 2023-2024. Returner JSON klar til bruk for /hentJSONTabellData, med kodetekst=J og sortering på Institusjonskode og Årstall.

Begrens til NTNU og UiO.

## Forslag til JSON spørring

```
{
  "tabell_id": 60,
  "api_versjon": 1,
  "groupBy": ["Institusjonskode", "Årstall"],
  "sortBy": ["Institusjonskode", "Årstall"],
  "filter": [

{"variabel": "Institusjonskode", "selection": {"filter": "item", "values": ["NTNU", "UiO"]}},

{"variabel": "Årstall", "selection": {"filter": "between", "values": ["2023", "2024"]}}
  ],
  "kodetekst": "J"
}
```

## cURL (valgfritt) basert på samme JSON

```
curl -X POST "https://dbh.hkdir.no/api/hentJSONTabellData" \
-H "Content-Type: application/json" \
-H "Authorization: Bearer $DBH_JWT" \
-d '{
  "tabell_id": 60,
  "api_versjon": 1,
  "groupBy": ["Institusjonskode", "Årstall"],
  "sortBy": ["Institusjonskode", "Årstall"],
  "filter": [

{"variabel": "Institusjonskode", "selection": {"filter": "item", "values": ["NTNU", "UiO"]}},

{"variabel": "Årstall", "selection": {"filter": "between", "values": ["2023", "2024"]}}
  ],
  "kodetekst": "J"
}'
```

Du kan nå legge resultat tabellen som kommer ut inn i GPT5 og be om visning av tabell og statistikkgraf for tabellen.

Husk å kontrollere resultatet nøye.

## Hente kanalregisterdata.

Velg tabell 851 for tidsskrift og 850 for forlag.

Eksemplene gjelder for tidsskrift men de samme prinsippene gjelder for forlag.

## Hente alle tidsskrift som er registrert eller endret for en bestemt dag:

```
{"tabell_id":851,"api_versjon":1,"statuslinje":"J","kodetekst":"J","desimal_separator":",",  
"variabler":["*"],"sortBy":[],  
"filter":[{"variabel": "Sist endret", "selection": {"filter": "like","values": ["%Jan 23 2026%"]}}]}
```

## For å hente alle tidsskrift med endringer mellom to datoer:

```
{"tabell_id":851,"api_versjon":1,"statuslinje":"J","kodetekst":"J","desimal_separator":",",  
"variabler":["*"],"sortBy":[],"filter":[{"variabel": "Sist endret","selection": {"filter":  
between", "values": ["Jan 01 2026", "Jan 23 2026"]}}]}
```

## Hente to tidsskrift(Du kan også søke på et. Tidsskrift er kommaseparert):

```
{"tabell_id":851,"api_versjon":1,"statuslinje":"J","kodetekst":"J","desimal_separator":",",  
"variabler":["*"],"sortBy":[],"filter":[{"variabel": "Original tittel","selection": {"filter": "item",  
"values": ["Nature","Applied Mathematics and Computation"]}}]}
```

## Hente alle tidsskrift:

```
{"tabell_id":851,"api_versjon":1,"statuslinje":"J","kodetekst":"J","desimal_separator":",",
```

```
"variabler":["*"],"sortBy":[],"filter":[{"variabel": "Original tittel","selection": {"filter": "all",  
"values": ["*"]}}]}
```

### Stier:

<https://kanalregister.hkdir.no/api/krtabeller/hentCSVTabellData>

**eller** /hentJSONTabellData for å hente JSON data.

### BULK

(for å hente hele tabellen): <https://kanalregister.hkdir.no/api/krtabeller//bulk-csv?rptNr=851>

BULK kan du kjøre direkte fra nettleser

### CURL script for å hente tidsskrift:

```
!/bin/bash
```

```
#Henter CSV:
```

```
curl -o ./resultCsv/851-result_csv.dat -d "@./queries/851-querybody.json" -H "Content-  
Type: application/json;charset=UTF-8" -v -X POST  
https://kanalregister.hkdir.no/api/krtabeller/hentCSVTabellData 2>&1
```

Det samme prinsippet gjelder for forlag.