

Direktoratet for høyere utdanning og kompetanse (HK-dir)

Database for statistikk om høyere utdanning (DBH) - API

Tilgang til data fra DBH data via programmerbart grensesnitt (API).

Database for høyere utdanning(DBH) tilbyr grensesnitt for å hente data om høyere utdanning programmatisk via API. Dette dokumentet er dokumentasjon for dette API'et.

04.05.2022 09:45

HK-dir Intern

Contents

Veiledning for spørringer mot API.....	2
Bruksvilkår	4
Registrering av bruker	5
Tolking av data	5
Endring av feltnavn og nye felt i tabellene.....	5
Innlogging og skjerming av data.....	5
DBH API klient.....	5
Eksempel på liste query.....	6
Eksempel med 'R' programmering.....	8
API-klient. Skjerm bilde.	8
Tabell	8
Spørring	8
Filter.....	9
Resultat.....	14
Statuskode.....	14
Teknisk informasjon for å hente data fra API via program - cUrl.....	15
cUrl: Eksempel på spørringen med en inputfil: querybody.json.....	15
Eksempel på AJAX query fra klient	16
Kodeverdier	17
Regler:.....	17
Last ned til katalog	18
Eksempel på spørring	19
BULK – Hente store datasett.	21
StreamCsvData – Hente store datasett.....	22
StreamJsonData – Hente store datasett.	22
Excel.....	23
Et steg videre.....	24
Dokumentasjon av tabeller og variabler.	25
Kunstig intelligens gjennom AI-agenter	26
Kort oppsummering.....	27
Sammenligning DBH Api og DBH Rapporter :.....	28
Dokumentasjon	29
Vanlige feilsituasjoner	30

Driftstabeller	30
Kort oppsummert	31

Veiledning for spørringer mot API

Database for høgre utdanning tilbyr statistikk API for spørringer mot variabler i databasen. Bruker kan bestemme selv hvilke data de vil hente.

API'et inneholder data organisasjoner, studietilbud, studentdata, doktorgradsdata, økonomidata, publiseringsdata fra kanalregisteret, personaldata og andre data(som for eksempel areal) . Dette utgjør en databank som du har tilgang til via api.

Det anbefales at brukere holder seg orientert om driftsendringer/nedetid gjennom å hente tabell 916-API-Driftsmeldinger eller gå inn på <https://dbh.hkdir.no/dbhapiklient/> og åpne knappen 'Meldinger'. Der vil det ligge dokumentasjon på nedetid og endringer i systemet.

Systemet omfatter datakilder som hver inneholder mange variabler og som kan sammenstilles av brukere av systemet. Brukere av systemet er i første rekke erfarne brukere som skal hente data, men det er også mulig å bruke og forstå for uerfarne brukere gjennom klient som er tilgjengelig via DBH sin nettside.

I tillegg kan data hentes via programmeringsspråk. For eksempel 'R', Python, Javascript, Java o.l.

Det er laget en egen syntaks på spørringen. Denne gjør at vi kan hente ut data effektivt. For å bruke systemet må brukere lære seg denne. Ta deretter 'copy' /'paste' av spørringen inn i ditt eget program/script og bruk eksempelet til å lage din egen spørring.

Spørringer mot DBH API er p.t tilgjengelig via HTTPS REST - POST og JSON format (JavaScript Object Notation).

Noen data er beskyttet etter personvern og GDPR regler(General Data Protection Regulation). Disse er merket vi tabell-1 (innholds tabell).

Det er valgfritt om brukere vil hente data via pålogging med token eller uten token. Hvis brukere henter persondata uten token, vil noen data være beskyttet av GDPR og avrundes til 0. Dette gjelder tall mindre enn 3. Pålogging med token er for institusjoner som er registrert hos HK-dir/DBH. Vi anbefaler bruk av token om du representerer en institusjon. Da vil du kunne se små data om egen institusjon.

Login fra eksterne maskiner gjøres via JSON Web Token (JWT). Bruker må selv lage rutiner for å hente nytt token før det gamle er gått ut på tid.

For å kunne sette opp gode spørringer, er det ønskelig å vite hvordan resultatet på spørringen blir. Dette kan vi se gjennom API klient som er laget. Formålet med klienten er å kunne bygge JSON/CSV spørringer på en enkel måte og kontrollere resultatet. Klienten kan også brukes til ad-hoc spørringer. Velge 'Tabell' fra øverste meny. Bruker justerer deretter spørringen og trykker de forskjellige knappene for spørring.

Klienten er knyttet opp mot beskrivelser av variabler som ligger på nettsiden til DBH.

Systemet produserer datafiler som blir sendt over Https. En datafil kommer som en funksjon av groupBy,variabler,sortBy og filter.

Variabler brukes på liste-tabeller som ikke inneholder groupBy. Her kan man velge variabler slik at man ikke trenger å hente alle feltene.

Gruppering og sortering er valgfrie felt men hvis det finnes grupperingsfelt og sorteringsfelt, må sorteringsfeltene være del av gruppering (slik som i SQL). Det er grupperingsfelt som kommer ut i spørringen. Som i SQL.

Alle tabeller må ha en forekomst i filterdelen av spørringen. Dette kan være for eksempel være 'all'.

En nærmere beskrivelse og eksempel på disse begrepene finnes i dokumentasjonen i dette dokumentet.

En nærmere beskrivelse av tabeller og variabler er gjort her:

<https://dbh.hkdir.no/datainnhold/tabell-dokumentasjon> .

Om man ønsker et mer lesbart format, kan man lagre filen lokalt og deretter laste inn i JSON eller CVS editor fra fil. Nederst på siden i klienten er det er link til en slik editor. Også andre kan brukes.

Data som genereres via API blir ikke lagret hos DBH. Alle forespørsler etter data blir logget. Logging av forespørsler blir utført for å kunne lage en statistikk som viser bruksmønster for datatjenesten. En slik statistikk kan være nyttig for å vurdere eventuelle forbedringer av tjenesten.

For å se om API kjører: <https://kanalregister.hkdir.no/api/Tabeller/> for kanalregister eller <https://dbh.hkdir.no/api/Tabeller/> for dbh.

Veier inn til DBH data via API.

1. Via klient – <https://dbh.hkdir.no/dbhapiklient> .
2. Programvare. Bruk av 'cUrl', 'Post', 'R', Javascript, GraphQL, Python, AI verktøy eller annen programvare.
Ved å bruke programvare, kan du hente flere tabeller samtidig og sette disse sammen til egne tabeller. Det er fire innganger fra programvarer.
 - Bulk. /bulk-csv. Get spørring. Vi henter hele tabellen og kan se den i nettleser.
 - StreamCsv/JSON. /streamCsvData/ streamJsonData. Post spørring. Utplukk av data med store datamengder.
 - Det er ikke mulig å bruke parameteret 'begrensning'. Forslaget: "begrensning":"10000",

er ment for hjelp til å unngå at nettleser blir fylt opp.

Sum antall skjerminger blir ikke levert på disse formatene.

hentCSVTabellData. /hentCSVTabellData. Post spørring. Mulighet for å bruke parameter 'begrensning'.

- hentJSONTabellData. /hentJSONTabellData. Post spørring.

Mulighet for å bruke parameter 'begrensning'.

Parameter begrensning kan brukes om det er ønskelig å hente data inn i skjermbilder.

Den tilsvarer sql begrepet 'top'. For eksempel top (1000).

Se også Tabell API innhold for å få oversikt over tabeller og tabeller med 'Bulk'.

Curl kan konverteres til programkode via ulike verktøy. For eksempel MuleSoft sin Advanced REST Client.

Vi tenker oss at når bruker skal sette opp en maskin til maskin kobling, så går han først inn i API Klienten på DBH sine nettsider og prøver ut spørringen. Da vil bruker få et eksempel i klienten som kan tilpasses.

Noen av variablene, for eksempel 'Institusjonskode' eller 'Avdelingskode', er DBH sine egne dataverdier og er ikke offisielle nasjonale koder.

Data leveres på UTF-8 karakter encoding.

Institusjonene bestemmer selv hvem som er brukere av institusjonen sine token. Riktig token gir tilgang til institusjonen sine data.

Vi anbefaler at dere oppgir e-mailadresse for at DBH kan komme i kontakt med dere om ulike driftssituasjoner.

Vi loggfører pålogging og henting av data i henhold til GDPR.

Lisens for data er: Norsk lisens for offentlige data (NLOD) 2.0. [Norsk lisens for offentlige data \(NLOD\)](#). HK-dir er ikke ansvarlig for konsekvenser som følge av feil i datagrunnlag eller system som ikke tilhører HK-dir.

Bruksvilkår

Vilkårene for bruk følger til enhver tid gjeldende Norsk lisens for offentlige data.

Legg spesielt merke til følgende:

HK-dir/DBH tar ikke ansvar for konsekvenser som følge av feil i systemet.

DBH 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 [Norsk lisens for offentlige data \(NLOD\)](#).

Registrering av bruker

Vi ønsker at aktive brukere av datatjenesten registrerer seg for bruk av api'et ved hjelp av brukertoken. Dette er for at vi bedre skal kunne skille på de forskjellige forespørslene som blir gjort, og kommunisere direkte med brukerne av data når det er behov for det.

E-postadresser blir ikke delt med noen og brukes kun for å verifisere brukere, generere API-nøkler og nå ut til brukerne av dataene.

Registrering er ikke obligatorisk.

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 enkel tabell og vi viser ellers til dokumentasjon for hver enkelt tabell og ellers til DBH sin dokumentasjon når det gjelder tolking av data.

Om man bruker AI verktøy slik som f.eks ChatGpt til analyse er det viktig å kontrollere analysen.

Endring av feltnavn og nye felt i tabellene

Vi vedlikeholder felt. Dokumentasjon på vedlikehold er i tabell for driftsmeldinger og i tabell 916-API-driftsmeldinger. Denne tabellen bør lastes ned for å følge med i endringer i systemet. Endringer kan innføres uten forvarsel selv om vi gir beskjed via mail når det er store endringer. Innføring av nye tabeller ligger også i 916 tabellen. Ved store endringer på tabellen kan det forekomme at vi legger inn en ny tabell.

Innlogging og skjerming av data

- Pålogging i API.
Vi har lagt om til innlogging via Feide fra 02.01.2023. Ta kontakt om dere ønsker å hente token.
- For tabeller som er merket for skjerming i 001-Innhold, vil persondata med verdier mindre enn 3 være anonymisert for brukere som ikke tilhører institusjonen.
- Om data er GDPR beskyttet, bør det fremkomme der de brukes hos bruker for å unngå mistolking av data.

DBH API klient

Dette er DBH sitt eget API verktøy som er nært knyttet til api'et.

Klienten bistår brukeren med å generere og redigere kode som kan benyttes for å hente data fra API.

Skjermbildet har informasjonssymbol med informasjon og regler for hver del av skjermbildet. De er nyttig å lese og inneholder det meste som skal til for å bruke systemet.

Bruker kan søke opp ønsket tabell og deretter gjøre sine tilpassinger på søket. Bruker kan velge antall poster og om kodetekster skal vise i resultatet av spørringen.

Vi viser en liste med variabler fra tabellen som bruker vil hente data fra. Der er link til beskrivelse av tabellen som vi ønsker å hente data fra under feltet for tabell.

Avsnitt for endepunkt viser hvor script kan hente data.

Bruker trykker "test spørring" og får tilbake JSON/CSV resultatsett. Spørring og resultatsett kan lastes ned lokalt til klienten, via knappen last ned ..., og deretter kan spørringen editeres/kopieres via Notepad++ eller en annen editor.

For å få resultatet mer lesbart kan data lastes ned lokalt med knapp 'Last ned'. Deretter kan datafilen hentes inn i editorene som det er link til nederst på klienten.

The screenshot shows the DBH API interface. At the top, there is a search bar for a table, currently showing '123-Registrerte studenter'. Below this is a list of variables with their corresponding formulas, such as 'Antall totalt' with the formula 'sum(Antall totalt)'. The 'Inngang' (Endpoint) section shows the URL 'https://api.nsd.no/dbhapi/Tabeller/'. The 'Spørring' (Query) section contains a JSON query:

```
{ "tabell_id": "123", "api_versjon": "1", "statuslinje": "N", "decimal_separator": ".", "groupBy": ["Institusjonskode", "Avdelingskode", "Årstall"], "sortBy": ["Institusjonskode", "Avdelingskode"], "filter": { "variabel": "Institusjonskode", "selection": { "filter": "all", "values": ["*"] } } }
```

. At the bottom, there are buttons for 'Test spørring - CSV', 'Test spørring - JSON', and 'Last ned'. Below the query editor, there is a section for 'Resultat (øverste del)' with a 'Last ned hele datasettet' button and options for 'CSV resultat', 'JSON resultat', and 'fra JSON til Excel'. A preview of the data is shown below these options.

Eksempelet viser et forslag til spørring med resultat.

Eksempel på liste query.

Være mest mulig presis når de henter data fra DBH-API'et.

`{"tabell_id":123,"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"]`, <-- Trenger ikke denne?

`"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.
    ]
  }
}
]]

```

Exclude trenger ikke legge ved, men kan brukes om noen verdier skal utelates.

```

"filter": "like",
"values": [
  "%Remote%"

```

% foran eller bak eller begge slik som vanlig sql.

```

{"tabell_id":123,"api_versjon":1,
"groupBy":["Institusjonskode", "Årstall"],
"sortBy":["Institusjonskode"],
"filter":[
  {
    "variabel": "Institusjonskode",
    "selection": {
      "filter": "all",
      "values": [
        "*"
      ]
    }
  }
  {
    "variabel": "Årstall",
    "selection": {
      "filter": "top", ← Bare ved årstall
      "values": [
        "1"
      ]
    }
  }
}
]]

```

Eksempel med 'R' programmering

Vi viser her til en egen klient på våre nettsider. <https://github.com/makinin/rdbhapi>

API-klient. Skjermbilde.

Tabell

I tabellfeltet søker vi frem den tabellen med tilhørende variabler som er aktuell for å hente data. Vi kan søke generisk eller spesifisert med tabellnummer. Hvis vi for eksempel skriver "studie", kommer alle tabeller som har "studie" i overskriften opp. Om vi skriver "11" kommer alle tabeller som inneholder 11 i tabellnummer eller navnet.

Noen tabeller er kodetabeller. Disse trenger datavarehuset for å bruke de vanlige datatabellene. De kan søkes opp med å skrive "Kodetabell" i søke-feltet for tabeller.

Ved å skrive liste kommer alle listetabeller frem.

Spørring

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 variabellisten. 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.

Viktige felt for spørring er:

- "tabell_id": Vi ser at det er samme tabell som vi søkte frem. Numerisk verdi.
- "api-versjon": Per dags dato er det bare 1 som er gyldig. Numerisk verdi.

- "groupBy/groupby": Dette er en liten tabell med variabler som starter med "[" og avslutter med "]".
Gruppering inneholder variabler. Her kan vi bestemme nivået på aggregeringen. Dette gjelder ikke for tabeller som er merket med liste. Disse kommer uten forslaget 'groupBy'. Forslaget starter med to nivå. Nivåene kan bruker bestemme og bestemmer dybden på tallene.
- sortBy/sortby: Her gjelder samme regler som i SQL.

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.

Filter

Eksempelene under er hentet fra skjermbildet i klienten og kan kopieres direkte. 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": ["xxx"] eller "exclude": ["xxx","xxx","xxx"]. Default er at feltet er tomt. "exclude": [""].

Eksempel:

```
{
  "variabel": "Tekst-egendefinert",
  "selection": {
    "filter": "all",
    "values": [
      "*"
    ]
  }
}
```

```

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

```

Her kan vi kopiere rett fra spørringen og inn i programmene til datavarehuset eller laste ned spørringen.

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

```

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

```

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

```

{
  "variabel": " Institusjonskode ",
  "selection": {
    "filter": "all",
    "values": [
      "*"
    ]
  }
},

```

item: Her henter vi alle forekomster som har verdier i listen slik som angitt i søket.

```

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

```

```
},
```

Items er adskilt med komma.

Vi kan også ha bare en verdi i rekken.

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

'between': Typisk intervall mellom årstall. Tilsvare *'between'* i sql.

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

'like': generisk søk. Tilsvare *'like'* i sql.

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

% kan være foran, bak eller begge steder. Vi kan også bruke exclude i filteret.

'lessthan': Typisk årstall eldre enn et gitt år, men kan også brukes på andre variabler

```
"filter": [
  {
    "variabel": "Institusjonskode",
    "selection": {
      "filter": "lessthan",
      "values": [
        "0500"
      ],
      "exclude": [
        "0264"
      ]
    }
  }
]
```

```
},
```

Dette eksempelet henter alle institusjoner som har institusjonskode mindre enn "0500", unntatt institusjonskode "0264".

For å unngå store datamengder, lønner det seg å sette opp spørringen med 'item' eller 'top'. Dette gjelder f.eks tabell for 900-studiepoeng, 208-emner og 347-studiepoeng. Hvis spørringen blir for stor, må den deles opp. Det blitt gitt melding om dette i meldingsfeltet.

En enkel egendefinert statistikk spørring:

```
{"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 egendefinert spørring med mange filter:

```
{"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"
      ]
    }
  }
]

```

Eventuelt kan spørringen deles og gjøres mindre:

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

Du kan legge inn begrensning i antall poster som blir returnert i klienten. '*' eller om du ikke har begrensning parameter gir alle poster.

Resultat

Hvis der er gruppering og summer er det bare de feltene som er i gruppering som vises sammen med sum gruppert. For tabeller med [] i gruppering, vises alle variablene slik som i SQL.

Første linjen i resultatsett er statuslinje for resultat i inngang. Linjen inneholder:

"tabell_id ": Identitet for tabell. Dokumentasjon om tabellen finnes ovenfor.
"api_versjon ": - P.t er det versjon 1.
"leveransenr ": Hver spørring får et unikt nummer som blir registrert hos DBH.
"leveringstid": Når data blir sendt.
"antall": Antall JSON poster funnet.
"kodetekst": For noen tabeller kan bruker bestemme at de vi se hva kodene betyr (j/n)
"begrensning": Antall JSON poster som ønskes returnert. '*' gir alle. 1000 er forslag.
"antall linjer anonymisert ": Poster skjermet etter personvernloven(GDPR).
"returkode ": 0 = Ok. Ev. andre koder. Foreløpig bare denne.
"melding ": "Data levert" ev. andre meldinger som kommer fra API.

Denne linjen er valgfri. "statuslinje":"N" kan endres til "J". Levers med /hentCSVTabellData
,/hentJSONTabellData.

Deretter kommer datadelen [{}].

Statuskode

Sammen med http metoder som API responderer til, vil det bli returnert http statuskoder.

Status 200 indikerer at request er ok. Status 204 betyr at request er ok, men ingen data finnes.

Returkoder i 400 segmentet, indikerer at noe gikk galt. Det kan være syntaks i JSON, Innhold i JSON eller at du prøver å gjøre noe du ikke er autorisert for. Dette er klientside problem.

Returkoder i 500 segmentet indikerer et server-side problem slik at forespørselen ikke kan utføres. Du får det når du ikke får kontakt med server.

Teknisk informasjon for å hente data fra API via program - cUrl

Illustrert med cUrl.

Det forventes at bruker behersker cUrl.

Returdata er JSON eller CSV, avhengig av om du velger å kalle /streamCsvData eller /streamJsonData. Det anbefales å bruke stream funksjonene ved større datamengder.

URL som du skal bruke, vil stå i konfigureringsbildet for API'et. Den brukes i forbindelse med https 'POST' spørring over linje til DBH.

Hent token(Hentes fra FEIDE):

Eksempel på script med nye adresser:

```
JWT=$(curl --data "grant_type=client_credentials" --user <feide-id><feide-pwd>:  
https://auth.dataporten.no/oauth/token | grep -Eo '"access_token"[^,]*' | grep -Eo '[:]*$' | sed  
's/[=]//g')  
echo $JWT  
#JSON  
curl -o ./resultJson/001-resultJson.dat -d "@./queries/001-querybody.json" -H "Content-Type:  
application/json;charset=UTF-8" --header "Authorization: Bearer $JWT" -v -X POST https://dbh-  
data.dataporten-api.no/Tabeller/hentJSONTabellData 2>&1
```

-i gir statusinformasjon og kan tas bort etter ønske.

cUrl: Eksempel på spørringen med en inputfil: querybody.json

Denne filen kopieres direkte fra spørringen som er lastet ned fra klienten .

Om det blir brukt Linux kan filen kopieres fra Windows til Linux. Tegnsett blir konvertert til Unix tegnsett om Unix blir brukt.

```
{"tabell_id":132,"api_versjon":1,"statuslinje":"N","kodetekst":"N","decimal_separator":".",  
"groupBy":["Institusjonskode","Avdelingskode","Årstall"],  
"sortBy":["Institusjonskode","Avdelingskode","Årstall"],  
"filter":[  
  {  
    "variabel": "Institusjonskode",
```

```

    "selection": {
      "filter": "all",
      "values": [
        "*"
      ]
    }
  },
  {
    "variabel": "Avdelingskode",
    "selection": {
      "filter": "all",
      "values": [
        "*"
      ]
    }
  },
  {
    "variabel": "Årstall",
    "selection": {
      "filter": "top",
      "values": [
        "2"
      ]
    }
  }
}
]]

```

JSON kan også legges som en variabel i programet.

Eksempel på AJAX query fra klient

(Representert ved javascript biblioteket AXIOS).

stitalapi = <sti til api>

jquery = innhold i spørring boksen i bildet.

```

let config = {
  onUploadProgress: function(progressEvent) {
    let percentCompleted = Math.round( (progressEvent.loaded * 100) / progressEvent.total );
  }, //progressindikator
  headers: {
    'Authorization': 'Bearer ' + this.props.hashToken,
  }
};

```

Spørring via Javascript - Axios:

```

Axios.post(stitalapi, {
  node: jquery
},config)
.catch(error => {
  if (error.response) {
    .....
  } else if (error.request) {
    .....
  } else {
    .....
  }
}

```

```
    }  
});
```

HTTP error vil vise i linjen for Statuskode i konsollet eller leses ut fra "error" seksjonen. I Axios er ikke \ nødvendig, slik som det er gjort i eksempelet nedenfor.

Kodeverdier

For utvalgte koder vil det vise utfyllende tekst ved å velge: "kodetekst":"J" i spørringen. Kodetekst er valgfri.

Spørringen vil gå litt lenger responstid med utfyllende navn på kodene. For store spørringer i nettleser, kan det lønne seg å dele opp spørringen. Du kan ikke søke frem data via kodetekster. Søk foregår via variabler som er angitt i variabel delen av bildet.

Regler:

For avdelingsnavn, må Institusjonskode være med i groupby setningen.

For studieprogramkode, må Institusjonskode, Årstall og Semester være med i groupBy. For noen tabeller (f.eks 335-Utdanningsplan, er ikke semester med).

For de andre variablene er det ikke slike hensyn.

I følgende variabler skal det være mulig å hente kodetekst/navn ved tabeller som har groupby.

Eksempel:

Spørring: ❸

```
{
  "tabell_id": 60,
  "api_versjon": 1,
  "statuslinje": "N",
  "kodetekst": "J",
  "decimal_separator": ".",
  "groupBy": ["Institusjonskode", "Avdelingskode", "Årstall", "Semester"],
  "sortBy": ["Institusjonskode", "Avdelingskode"],
  "filter": [
    {
      "variabel": "Institusjonskode",
      "selection": {
        "filter": "all",
        "values": [
          "sg"
        ]
      }
    }
  ]
}
```

Resultat (øverste del): ❸

Last ned hele datasettet:

CSV resultat

JSON resultat

fra JSON til Excel

```
[
  {
    "Institusjonskode": "0217",
    "Institusjonsnavn": "Samisk høgskole",
    "Avdelingskode": "000000",
    "Avdelingsnavn": "SH (uspesifisert underenhet)",
    "Årstall": "2018",
    "Semester": "Vår",
    "Antall totalt": "0",
    "Antall kvinner": "0",
    "Antall menn": "0"
  },
  {
    "Institusjonskode": "0217",
    "Institusjonsnavn": "Samisk høgskole",
    "Avdelingskode": "000000",
    "Avdelingsnavn": "SH (uspesifisert underenhet)",
    "Årstall": "2017",
    "Semester": "Høst",
    "Antall totalt": "0",
    "Antall kvinner": "0",
    "Antall menn": "0"
  },
]
```

Last ned til katalog

Test spørring som CSV:

- Last ned fil
- Åpne Excel og importer fil.
- Trykk ferdig.

Test spørring som JSON:

- Last ned fil.
- Åpne med notepad++ eller lignende editor.

Eksempel på spørring

Detaljert egendefinert statistikk spørring:

```
{
  "tabell_id":123,"api_versjon":1,"statuslinje":"N","decimal_separator":".",
  "groupBy":["Institusjonskode",
  "Årstall","Avdelingskode","Semester","Studentkategori","Studieprogramkode"],
  "sortBy":["Institusjonskode",
  "Årstall","Avdelingskode","Semester","Studentkategori","Studieprogramkode"],
  "filter":[
    {
      "variabel": "Institusjonskode",
      "selection": {
        "filter": "item",
        "values": [
          "1150"
        ]
      }
    },
    {
      "variabel": "Avdelingskode",
      "selection": {
        "filter": "between",
        "values": [
          "000000","230000"
        ]
      }
    },
    {
      "variabel": "Årstall",
      "selection": {
        "filter": "top",
        "values": [
          "2"
        ]
      }
    },
    {
      "variabel": "Semester",
      "selection": {
        "filter": "greaterthan",
        "values": [
          "1"
        ]
      }
    }
  ]
}
```

```
}  
}  
]}
```

Egendefinert spørring:

```
{"tabell_id":226,"api_versjon":1,"statuslinje":"N","decimal_separator":".",  
"groupBy":["Institusjonskode", "Årstall","Adresse/Navn","Leieforhold","Når avtalen opphører"],  
"sortBy":["Institusjonskode", "Årstall","Adresse/Navn","Leieforhold","Når avtalen opphører"],  
"filter":[  
  {  
    "variabel": "Institusjonskode",  
    "selection": {  
      "filter": "item",  
      "values": [  
        "1150"  
      ]  
    }  
  },  
  {  
    "variabel": "Årstall",  
    "selection": {  
      "filter": "top",  
      "values": [  
        "10"  
      ]  
    }  
  },  
  {  
    "variabel": "Adresse/Navn",  
    "selection": {  
      "filter": "item",  
      "values": [  
        "Biologisk stasjon"  
      ]  
    }  
  }  
]}
```

Eller å undersøke kvinneandel på avdelinger i 2018:

Denne spørringen gir antall årsverk for stillingskodene 1013, 1017, 1352, 1378 og 1404 ved NTNUs fakultet for informasjonsteknologi og elektroteknikk:

```
{"tabell_id":225,"api_versjon":1,"statuslinje":"N","begrensning":"1000","kodetekst":"J","decimal_se  
parator":".",  
"groupBy":["Institusjonskode", "Avdelingskode", "Stillingskode", "Årstall"],  
"sortBy":["Institusjonskode", "Avdelingskode"],  
"filter":[  
  {  
    "variabel": "Institusjonskode",
```

```

"selection": {
  "filter": "item",
  "values": [
    "1150"
  ]
},
{
  "variabel": "Avdelingskode",
  "selection": {
    "filter": "item",
    "values": [
      "273000","273100","273321","273323","273324","273325","273326","273327","273328","273823",
      "273824","273840","273850"
    ]
  },
  {
    "variabel": "Stillingskode",
    "selection": {
      "filter": "item",
      "values": [
        "1013","1017","1352","1378","1404"
      ],
      "exclude": [
        ""
      ]
    }
  },
  {
    "variabel": "Årstall",
    "selection": {
      "filter": "item",
      "values": [
        "2018"
      ]
    }
  }
}
]]

```

BULK – Hente store datasett.

Dette er en enkel 'HTTPS GET' kommando for åpne data. D.v.s data som ikke er beskyttet av GDPR.

Fra nettleser skriver vi:

<https://dbh.hkdir.no/api/Tabeller/bulk-csv?rptNr=001>

I cUrl skrives det f.eks:

```
curl -o ./ResultBulk/001.csv -H "Content-Type: application/json;charset=UTF-8" -v -X GET https://dbh.hkdir.no/api/Tabeller/bulk-csv?rptNr=001 2>&1
```

Tabellen blir streamet ned lokalt. Det er ikke mulig å legge inn andre kriterier enn tabellnr og du henter hele tabellen.

De tabellene som kan lastes ned er merket som bulktabeller i 001-API-Innhold.

Det er ikke mulig å hente datatabeller som beskyttes av gdpr loven.

StreamCsvData – Hente store datasett.

Dette kan sees på som en fortsettelse eller erstatning for BULK. Her er det mulig å laste ned store datamengder for alle tabeller med bruk av queries.

Dette kan også gjøres i API klienten.

Eksempel:

```
JWT=$(curl --data "grant_type=client_credentials" --user <feide-id><feide-pwd>: https://auth.dataporten.no/oauth/token | grep -Eo ""access_token"[\^,]*" | grep -Eo '[\^:]*$' | sed 's/[=]//g')
echo $JWT
#JSON
curl -o ./resultJson/001-resultJson.dat -d "@./queries/001-querybody.json" -H "Content-Type: application/json;charset=UTF-8" --header "Authorization: Bearer $JWT" -v -X POST https://dbh-data.dataporten-api.no/Tabeller/ streamCsvData 2>&1
```

Data vil bli beskyttet etter GDPR loven.

Data vil måtte importeres i Excel og tilpasses tegnsatt Excel manuelt.

StreamJsonData – Hente store datasett.

Dette kan sees på som en fortsettelse eller erstatning for BULK. Her er det mulig å laste ned store datamengder for alle tabeller med bruk av queries.

Dette kan også gjøres i API klienten. Da kan resultatet hentes inn i en nettleser f.eks Firefox.

Eksempel:

```
JWT=$(curl --data "grant_type=client_credentials" --user <feide-id><feide-pwd>: https://auth.dataporten.no/oauth/token | grep -Eo ""access_token"[\^,]*" | grep -Eo '[\^:]*$' | sed 's/[=]//g')
echo $JWT
#JSON
curl -o ./resultJson/001-resultJson.dat -d "@./queries/001-querybody.json" -H "Content-Type: application/json;charset=UTF-8" --header "Authorization: Bearer $JWT" -v -X POST https://dbh-data.dataporten-api.no/Tabeller/ streamJsonData 2>&1
```

Data vil bli beskyttet etter GDPR loven.
I dette datasettet vil numeriske tall være representert som numeriske.

Excel.

Vi antar at brukere av systemet kjenner litt til Excel.

Importen til Excel vil bli bestemt av hvordan Excel er satt opp lokalt. Problemer kan være knyttet til norsk tegnsett, decimaltegn og datoformat.

Noen løsninger:

Import fra 'Last ned CSV resultat'.

Velge 'Data' – 'Fra tekst' – 'Importer' – 'Data med skilletegn' – 'Filopprinnelse: Unicode (Utf-8)'
'Skilletegn – Semikolon'. Dette er standard metoden for import av datafilen.

Ved å velge importer som tekst på feltnivå, vil feltet vise slik som i datafilen.

Et steg videre

Eksempel på spørring:

DBH statistikkbank

DATABASE FOR STATISTIKK OM HØYERE UTDANNING
API konfigurering og spørring

Logg inn
Institusjonskode: 0000
Api status: "Aktiv"

Tabell: 135-Utenlandske studenter

Beskrivelse: <https://dbh.hkdir.no/datanhold/tabell-dokumentasjon/135>

Variabler:

Variabel	Formel
Institusjonskode	
Avdelingskode	
Årstall	
Semester	
Studieprogramkode	
Landkode	
Antall totalt	sum(Antall totalt)
Antall kvinner	sum(Antall kvinner)
Antall menn	sum(Antall menn)

Variabler fra studieprogrammet:	
Studiumkode	
Nivåkode	
Andel av heltid	
Studiepoeng	
Prosent egenfinansiering	
NUS-kode	
Organisering_kode	
Andel praksis	
Inderspråk	

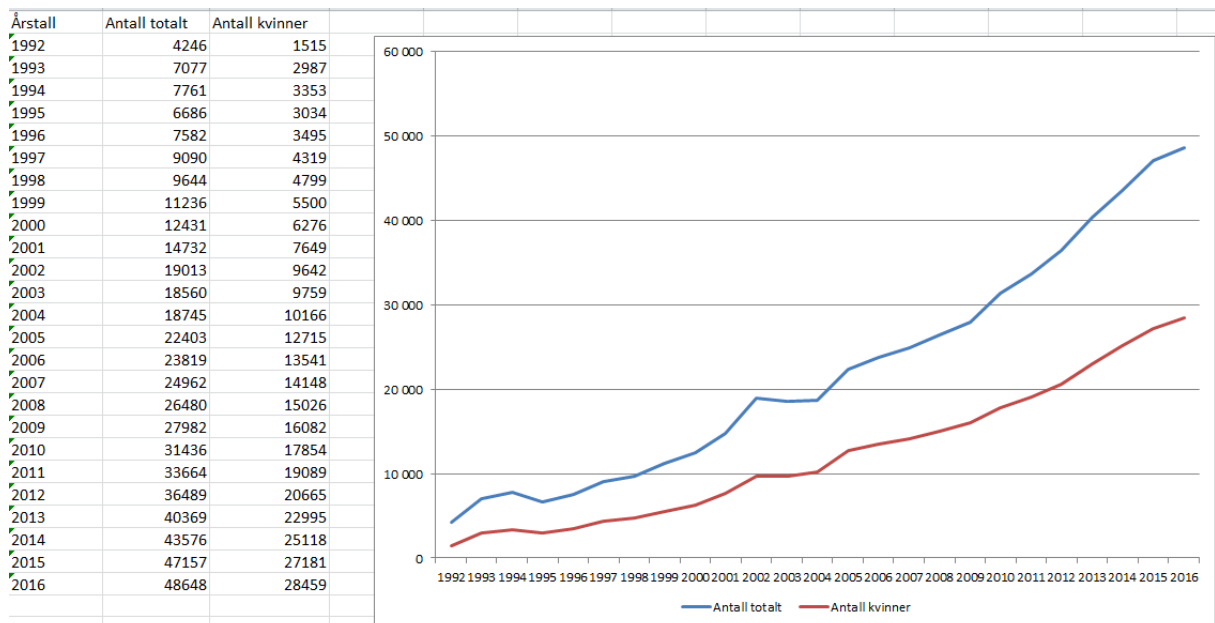
Endepunkt: DBH data: <https://dbh.hkdir.no/api/Tabeller/>
Kanalregister data (850-854): <https://kanalregister.hkdir.no/api/Tabeller/>
Mindre datasett: /henkCSVTabellData eller /henkJSONTabellData
Stream datasett: /streamCsvData eller /streamJsonData
Bulk datasett: /bulk-csv/?pnr=???

Spørring:

```
filter : [
  {
    "variabel": "Årstall",
    "selvskjutt": {
      "filter": "all",
      "values": [
        "2020"
      ]
    }
  }
]
```

Direkte - JSON Direkte - CSV Spørring - CSV Spørring - JSON Lagre spørring

Disse dataene kan vi legge over i Excel.



Her viser vi tabell for utenlandske kvinnelige studenter fra 1992 til 2016.

Dokumentasjon av tabeller og variabler.

Dokumentasjon av tabeller kan hentes slik:

```
{"tabell_id":913,"api_versjon":1,"statuslinje":"N","begrensning":"*","kodetekst":"J","desimal_separatør":",",  
"variabler":["*"]}
```

```

"sortBy":["Emne id", "Tabell id"],
"filter":[
  {
    "variabel": "Tabell id",
    "selection": {
      "filter": "item",
      "values": [
        "60"
      ]
    }
  }
]

```

For hver tabellen sine variabler, må man lese tabell 2-metadatas.

Dokumentasjonen ligger med HTML og andre formaterings tegn til hjelp for bruker.

Dokumentasjon for det enkelte felt ligger på 002-metadatas. Du kan for eksempel bare hente de feltene som er integer eller decimal og derved få informasjon om disse typene. Anta da at de andre er char felter. F.eks:

```

{"tabell_id":2,"api_versjon":1,"statuslinje":"N","begrensning":"100","kodetekst":"N","desimal_separ
ator":".",
"variabler":["Variabel navn","Datatype"],
"sortBy":["Tabell id", "Datalengde"],
"filter":[
  {
    "variabel": "Tabell id",
    "selection": {
      "filter": "item",
      "values": [
        "112"
      ]
    }
  },
  {
    "variabel": "Datatype",
    "selection": {
      "filter": "item",
      "values": [
        "int","decimal"
      ]
    }
  }
]

```

Metadatas kan legges inn i minnetabell. Da slipper man flere oppslag.

Kunstig intelligens gjennom AI-agenter

For å bygge en enkel AI-agent for en tabell/datasett kan vi bruke en generell modell(f.eks ChatGpt).

1. **Importere datasettet til et analysemiljø. For eksempel ChatGpt.**

2. **Lage funksjoner for spesifikke forespørsler og analyser. Eller spør direkte i ChatGpt.**
3. **Brukergrensesnitt:** Et enkelt tekstbasert grensesnitt for å stille spørsmål til AI-agenten.

Agenten kan da automatisk generere tabeller og grafer med trender over tid.

Husk å kontrollere nøye at resultatet er rett. GDPR regler må følges slik at små data om egen institusjon ikke blir lagret i AI.

Noen agenter f.eks i ChatGpt vil også kunne generere et forslag til Pythonkode som kan brukes som utgangspunkt til ulike formål.

DBH api bruker POST spørringer. Dette har betydning for hvordan AI agenter utvikles.

Kort oppsummering

API omfatter en klient som kan brukes både av utviklere for å inkludere spørringer i egne program og vanlige statistikere som kan hente ut data og laste disse over i Excel for videre behandling.

Data blir levert i to format – CSV og JSON. CSV gir mindre filer og i noen tilfeller lønner det seg å hente data som CSV fil.

Det er mulig å ta 'Copy/Paste' av spørringen over i andre program og deretter endre verdier programmatisk og bruke dette som fast input i datavarehus og andre programsystemer.

For datavarehus antar vi at utviklere kjenner til JSON, CSV og cUrl.

I noen tilfeller har nettleser problemer med store datamengder. Da lønner det seg å hente data via CSV. Filter 'all' kan endres til 'item', 'greaterthan' eller 'between' for å få mindre utvalg. Bruker kan også bruke stream funksjonen og laste data direkte til fil.

Ved eventuelle feil og mangler kan DBH kontaktes.

Kilder:

<https://www.ssb.no/omssb/tjenester-og-verktoy/api/px-api>

Her ligger også instruksjonsvideo og eksempel på programmering i ulike språk.

Autorisering skjer pr. type pålogging. For eksempel må DBH ha beskjed om at du trenger autorisering om du logger inn som DBH bruker og trenger autorisering til å se data for egen institusjon, selv om du har gyldig autorisering som en annen type pålogging.

Tips og triks:

- Du trenger ikke alltid bruke "variabler":["*"] for lister. Du kan oppgi hvilke felt du ønsker slik at du ikke trenger å hente alle feltene i en tabell.

- MS Explorer. I noen versjoner virker ikke 'fra JSON til Excel'. Bruke Spørring – CSV og importerer .txt fil i Excel.
- Angi tilstrekkelig antall posisjoner i søket. F.eks må institusjonsnr ha fire siffer. «123» vil ikke gi tilslag. «1234» vil gi tilslag.
- Store datamengder i klient: Nettleser kan gå tom for minne. Tøm minnet og eventuelt dele opp søket (f.eks bruke 'between' eller 'item'). Du vil ikke få samme problemene om du bruker cUrl script. Du kan også bruke stream funksjonene.
- Du definerer deg som bruker ved å logge inn i API på klient. I tillegg må DBH autorisere deg for å se alle data om egen institusjon. Ta kontakt med DBH om dette.
- I noen rapporter: (347,208,900,901,905,701...) må man velge 'item' og skrive inn institusjonsnr i klient. "<Skriv inn institusjonsnr>". Dette trenger du ikke hvis du henter via cUrl.
-
- Du sparer ca. 50-70 % filstørrelse(avhengig av antall felt) om du henter i filformat CSV. Dette filformatet anbefales for store datamengder.
- Om du ønsker dypere statistikk enn tre nivå kan du kopiere variabler inn i groupby fra 'Variabler' delen av bildet.
- Applikasjonen kan også kjøres på nettbrett. Bruker henter data som CSV. Laster ned fil og klikker på nedlastingsfilen. Denne skal da komme opp i Excel. Kan så bruke Excel til å søke i filen.
- Ved download av csv format og problemer norsk tegnsett. Stå i Excel. Velge 'Data' – 'Fra tekst' – 'Importer' – 'Data med skilletegn' – 'Filopprikkelse: Unicode (Utf-8)' 'Skilletegn – Semikolon'. Da skal Excel vise norske tegn.

Sammenligning DBH Api og DBH Rapporter :

I noen tilfeller ser vi at det er differanser mellom summene som vises i DBH rapportdel og de summene som vises i API.

De vanligste årsakene er:

- Valg/avkryssinger i DBH rapport.
I API/Klient henter bruker ofte hele tabellen hun/han ønsker å se.
I rapportdelen gjøres valg som fører til at data blir filtrert bort. Derfor kan det oppstå større differanser.
I API oppretter vi et forslag til spørring som bruker kan endre og legge inn nye filter.
Forslaget er basert på at alle data hentes. Bruker kan selv legge inn begrensninger på data som skal leveres.

- Valg av avrunding i DBH rapporter.
Av og til ser vi differanser på 1(f.eks) mellom API og DBH Rapporter.
Dette skyldes mest sannsynlig avrunding av data i Rapportdelen.
I API avrunder vi ikke data.
Om du er pålogget på Statstikk siden kan du velge å ta bort avrunding.
Da vil små differanser vanligvis forsvinne.
- Pålogging i API.
Om du ikke er definert for å se persondata (små tall) for egen institusjon, må du opprette bruker. Velg Logg inn og opprett bruker i API klient bildet.
Send deretter beskjed til DBH om at vi må opprette autorisering til å se små tall
Logg deretter inn og spør på nytt.
Om nødvendig kan du teste med å sette "statuslinje":"J". Da vil du få beskjed om antall skjerminger.

Dokumentasjon

Om bruker kjører ut tabell 913-Dokumentasjon via 'Direkte – Json' knapp vil han få en liste som kan hentes opp i nettleser:

```

{
  "0": {
    "Emne id": "1",
    "Emne": "Ikke tilordnet emne",
    "Tabell id": "1",
    "Tabellnavn": "API innhold",
    "Bruk av data": "Oversikt over tabeller og variabler",
    "Definisjon": "Oversikt over hvilken tabell i API som inneholder variabler fra DBH.",
    "Beskrivelse": "API innhold"
  },
  "1": {
    "Emne id": "1",
    "Emne": "Ikke tilordnet emne",
    "Tabell id": "2",
    "Tabellnavn": "API metadata",
    "Bruk av data": "Oversikt over metadata",
    "Definisjon": "Oversikt over metadata",
    "Beskrivelse": "API metadata"
  },
  "2": {
    "Emne id": "1",
    "Emne": "Studentdata",
    "Tabell id": "68",
    "Tabellnavn": "Studenter fordelt på alder",
    "Bruk av data": "Studenter fordelt på alder rapporteres i DBH som en oversikt over antall studenter fordelt på årskull (antall 20-åring, 21-åring etc.) og aldersintervaller.",
    "Definisjon": "Dette er antall studenter fordelt på alder på grunnlag av fødselsdato. Studentenes aldersfordeling rapporteres på det enkelte studieprogram. En student skal telle med kun på ett program, jmf. tabellen REGISTRERTE STUDENTER. Data omfatter alle studentkategoriene.",
    "Beskrivelse": ""
  },
  "3": {
    "Emne id": "1",
    "Emne": "Studentdata",
    "Tabell id": "62",
    "Tabellnavn": "Utvekslingsavtaler",
    "Bruk av data": "",
    "Definisjon": "",
    "Beskrivelse": ""
  },
  "4": {
    "Emne id": "1",
    "Emne": "Studentdata",
    "Tabell id": "66",
    "Tabellnavn": "Desentralisering og fjernundervisning",
    "Bruk av data": "Data for denne tabellen benyttes for Tabell 1.11 i UFD's budsjetttrunkskriv. Data benyttes for dekning av merkostnader ved internt finansierte desentraliserte studietilbud, som bevilges over UFDs budsjettkapitler.",
    "Definisjon": "Her rapporteres studentdata på fleksible utdanninger, dvs. desentraliserte tilbud og fjernundervisningstilbud. Med desentralisering eller fjernundervisning forstås undervisning som geografisk er lokalisert andre steder enn ved Institusjonenes ordinære studiesteder. I de tilfelle der en avdeling har et tilbud på et sted der en annen avdeling ved samme institusjon er lokalisert, så skal ikke dette regnes som et desentralisert tilbud, og ikke rapporteres i denne tabellen. Studenter som omfattes av en samarbeidsavtale med andre institusjoner innen høyere utdanning skal heller ikke rapporteres.",
    "Beskrivelse": "Dersom et nytt studieprogram skal oppstartes det kommende år, dvs. at det ikke finnes faktisk registrerte studenter på programmet, men kun prognosetall for neste år, tas disse programmene med, og da med angivelse av prognosetallene. Videre, dersom det for rapporteringsåret finnes registrerte studenter på et studieprogram som er tatt opp ulike år, f.eks. både i 2001 og 2002, rapporteres dette som to rekker i tabellen."
  },
  "5": {
    "Emne id": "1",
    "Emne": "Studentdata"
  }
}

```

Vanlige feilsituasjoner

Ut fra vår logg ser vi noen vanlige feil som er enkle å rette:

- Kodetekst må stå til 'J' for at forklarende tekst skal komme frem.
- Det er en grense på 1 million poster uthenting. http returkode 304.
- Studieprogramnavn kan bare vise når årstall, semester og studieprogramkode er med i querien.
- 854-Tidsskrift – forgjengere og etterfølgere.
Bytt ut: "sortBy":["Forgjenger id","Etterfølger id"], med
"sortBy":["Forgjenger id"],
- Kjøring fra script. Ikke kopier spørring fra network i nettleser. Kopier spørring fra Spørring i Api-klient. Alså uten {"node":{ }.
- Format på dato:
- 850-Publiseringskanaler.

```
{  
  "variabel": "Sist endret",  
  "selection": {  
    "filter": "between",  
    "values": [  
      "Apr 15 2020", → F.eks apr. 15 2020 er feil format  
      "Apr 29 2020",  
    ],  
    "exclude": [  
      ""  
    ]  
  }  
}
```

I noen tilfeller kan få differanser mellom API tabeller og DBH rapporter. Når vi setter opp rapportdelen med riktige avkryssinger i forhold til API, vil summene være de samme. For eksempel høyskoler som er standard med i Kunnskapsdepartementet(KD) strukturen i rapportdelen, men som ikke tilhører KD (politi og militære skoler).

Ta utgangspunkt i forslaget i API-klienten ved design av spørring.

Driftstabeller

Tabbel-001-innhold, og tabell 002-API-metadata gir beskrivelse av tabeller.
Tabell 913-917 gir nærmere beskrivelse av driftsmeldinger og endringer i felt og tabeller.

Eksempel på script som henter driftsmeldinger for september 2020.

```
{"tabell_id":916,"api_versjon":1,  
"variabler":["*"],  
"sortBy":["Dato"],  
"filter":[
```

```

{
  "variabel": "Dato",
  "selection": {
    "filter": "like",
    "values": [
      "2020-09%"
    ]
  }
}
]]}
Eller en periode:
{"tabell_id":916,"api_versjon":1,
"variabler":["*"],
"sortBy":["Dato"],
"filter":[
  {
    "variabel": "Dato",
    "selection": {
      "filter": "between",
      "values": [
        "2020-09-01","2020-10-30"
      ]
    }
  }
]}

```

Kort oppsummert

API'et leverer data basert på det spørrespråket ber om. Det betyr at det kan oppstå feil om data blir sammenstilt feil. For eksempel kan du få for høye tall om du ukritisk summerer innenfor årstall for studenter. Da kan du sammenligne med rapportdelen av DBH og se om tallene er like. I rapportdelen vil dere få feilmelding om felt er obligatoriske.

Funksjoner for å hente data.

- Strømme data. Mulighet for å hente store datamengder ved Bulk, /streamCsvData eller /streamJsonData.

For mindre datamengder og on-line spørringer mot API, kan /hentCSVTabellData eller /hentJSONTabellData brukes.

Utplukk av data.

Verb som kan brukes til å plukke data: greaterthan, lessthan, between, like, item, all, top, exclude, include.

For eksempel:

```

{
  "variabel": "Utgiver",
  "selection": {
    "filter": "like",
    "values": [
      "%MDPI%"
    ]
  }
}

```

Saman med "begrensning":"1", gir dette den øverste på listen som er spesifisert i sortering.
 Eller:

```

{
  "variabel": "Årstall",
  "selection": {
    "filter": "between",
    "values": [
      "2016","2020"
    ],
    "exclude": [
      "2017"
    ]
  }
}

```

Variabler

Det er også mulig å velge ut hvilke variabler som skal levere data i spørringen.
 For tabeller med groupby, vil groupby bestemme hvilke variabler som blir sendt.
 For tabeller uten groupby(lister), kan bruker skrive inn hvilke variabler som skal levere data.
 Forhåndsverdi er '*'. Det vil si alle variabler i tabellen. Dette kan endres av bruker til færre variabler enn det som står i listen over variabler for tabellen.

Begrensing.

Forhåndsverdi er 10000 men bruker kan fritt velge verdi.
 Vi anbefaler at her står '*' eller at denne parameteren tas bort om bruker skal ha ut alle data som er større enn 10000 linjer med data.
 Meningen er at denne kunne brukes for å holde kontroll på data som ble sendt til nettleser, slik at bruker kunne teste spørringen uten at minnet til nettleser blir fylt opp.
 Bruker må vurdere om bruken av denne funksjonen.