

DBH-API

Tilgang til data via API

Database for høgre utdanning tilbyr API for spørringer mot variabler i databasen. Dette dokumentet viser hvordan API'et kan brukes.

Innhold

Veiledning for spørringer mot API.....	1
DBH API klient.....	3
Eksempel med 'R' programmering.....	4
Tabell.....	4
Spørring.....	5
Filter.....	6
Resultat.....	9
Statuskode.....	9
Teknisk informasjon for å hente data fra API via program - cUrl.....	10
Eksempel på AJAX query fra klient.....	11
cUrl: Eksempel på spørringen med en inputfil: querybody.json.....	11
Last ned til katalog.....	12
Eksempel på spørring.....	13
Et steg videre.....	15
Kort oppsummering.....	16

Veiledning for spørringer mot API

Database for høgre utdanning tilbyr API for spørringer mot variabler i databasen. Data er organisasjonsdata, studietilbud, studentdata, doktorgradsdata, økonomidata, publiseringsdata fra kanalregisteret, personaldata og andre data(som for eksempel areal) . Dette utgjør en statistikkbank.

Systemet omfatter litt under 100 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 klienten.

Det er laget en egen syntaks på spørringen. Denne gjør at vi kan hente ut data effektivt. For å bruke systemet, må bruker lære seg denne. Ta deretter copy /paste av spørringen inn i ditt eget program.

Spørringer mot DBH API er p.t tilgjengelig via Http REST - POST og JSON format (JavaScript Object Notation). Du kan endre inputverdiene ('values') programmatisk.

Data er beskyttet etter GDPR regler(General Data Protection Regulation). Det er valgfritt om bruker vil hente data via pålogging med token eller uten token. Hvis bruker henter persondata uten token, vil data være beskyttet av GDPR og avrundes til 0. Gjelder tall mindre enn 3.

Ved innlogging med token må bruker opprette brukerkonto selv. Da er bruker *autentisert* for systemet. Data vil fortsatt være skjermet etter personvernloven(GDPR). Hvis bruker er *autorisert* via

egen institusjon kan han se alle data fra egen institusjon, samt fra institusjoner som er slått sammen med egen institusjon. Ta kontakt med DBH om du ønsker å bli *autorisert*.

Logon til nettleser varer 12 timer. Deretter vil bruker bli logget av automatisk.

Logon fra maskin software via Json Web Token (JWT) vil vare i 1 time. 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 en klient som er laget. Formålet med klienten er å kunne bygge JSON/CSV spørringer på en enkel måte og kontrollere resultatet.

Velge 'Tabell' fra øverste nedtrekksmeny.

Bruker justerer deretter spørringen og trykker 'Test spørring'.

I klienten setter vi opp spørringen slik den blir på JSON format og kan kontrollere hva som kommer tilbake. Bruker kan tilpasse spørringen til sine behov og deretter legge spørringene inn i sine systemer.

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

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

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

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

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

En nærmere beskrivelse av tabeller og variabler er gjort her:

<http://dbh.nsd.uib.no/dokumentasjon/tabeller.action>.

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.

Data som genereres via API blir ikke lagret hos DBH.

Det er flere veier inn til DBH data via API.

1. Via klient – <https://api.nsd.no/dbhapiklient/>
2. Programvare. Bruk av 'cUrl', 'Post', 'R', Javascript eller annen programvare.

Vi tenker oss at når bruker skal sette opp en maskin til maskin kobling, så går han først inn i API klienten og prøver ut spørringen. Deretter kan han prøve spørringen i cUrl og til sist kan han programmere inn i egne program. cUrl finnes på de fleste Linux maskiner.

Hvis spørringen blir lastet ned på Windows maskin og deretter overført til Linux, må tegnsattet konverteres (f.eks Dos2Unix ev. Unix2Dos). Vi ser feil ved tegnsett hvis æøå er feil. For eksempel Å...rstell. Linjeskift er også forskjellig fra dos til unix format.

Systemet er et 'plug and play' system og det er laget en 'toolboks' (ligger på DBH sine nettsider) med eksempel på queries mot de fleste tabellene.

DBH API klient

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.

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.

Inngang viser hvor datavarehus kan hente data(URL).

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 client interface. It includes a search bar for tables, a list of variables with their formulas, an input field for the query, and a results section with download options for CSV, JSON, and Excel.

Tabell: 123-Registrerte studenter

Beskrivelse: <http://dbh.nsd.uib.no/dokumentasjon/tabell.action?tabellid=123>

Variabel	Formel
Institusjonskode	
Avdelingskode	
Årstart	
Semester	
Studenterkategori	
Studieprogramkode	
Antall totalt	sum(Antall totalt)
Antall kvinner	sum(Antall kvinner)
Antall menn	sum(Antall menn)

Inngang: <https://api.nsd.net/dbhapi/tjener/Tabeller/>
Velg mellom: /hentCSVTabellData eller /hentJSONTabellData

Spørring:

```
{
  "tabell_id":123,"api_versjon":1,"statuslinje":"","decimal_separator":",",
  "groupBy":["Institusjonskode", "Avdelingskode", "Årstart"],
  "sortBy":["Institusjonskode", "Avdelingskode"],
  "filter":{
    {
      "variabel": "Institusjonskode",
      "selection": {
        "filter": "all",
        "values": [
          "480000"
        ]
      }
    }
  }
}
```

Resultat (øverste del): Last ned hele datasettet:

CSV resultat | JSON resultat | fra JSON til Excel

```
"Institusjonskode";"Avdelingskode";"Årstart";"Antall totalt";"Antall kvinner";"Antall menn"
"0217";"000000";"2017";"0";"0";"0"
"0217";"000000";"2018";"4";"0";"0"
"0217";"400000";"2017";"154";"119";"35"
"0217";"470000";"2017";"67";"42";"25"
"0217";"480000";"2017";"201";"173";"28"
"0217";"510000";"2018";"271";"229";"42"
"0217";"520000";"2018";"222";"171";"51"
"0232";"440000";"2017";"1704";"1420";"275"
"0232";"440000";"2018";"1710";"1423";"287"
"0232";"461000";"2017";"937";"350";"507"
```

Eksempelet viser et forslag til spørring med resultat.

Eksempel med 'R' programmering

Dette eksempelet bruker ikke token pålogging.

```
#Eksempel med httr
```

```
library(foreign)
library(jsonlite)
library(httr)
```

```
library(rjstat)
```

```
options(encoding="UTF-8")
```

```
#Henter query
setwd(file.path("C:/Sync/Fak/2018/APler/queryBody"))
```

```
resultat<- POST(url="https://api.nsd.no/dbhapitjener/Tabeller/hentJSONTabellData",
  add_headers("Content-Type: application/json"),
  body=upload_file('060-querybody2.json'),
  encode='json')
```

```
resultat_data <- fromJSON(content(resultat, "text"))
```

```
setwd(file.path("C:/Sync/Fak/2018/APler/Result"))
```

```
#Fjerner linje 1 i data som inneholder Statuslinje
resultat_data = resultat_data[-1,]
```

```
#Fjerner alle variable med status i navnet
#resultat_data_clean<-resultat_data[, !names(resultat_data) %in% c("status")]
```

```
#Skriver til en csv-fil
write.csv(resultat_data_clean, file = "Tabell_060.csv")
```

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 tabellnr. Hvis vi for eksempel skriver "studie", kommer alle tabeller som har "studie" i overskriften opp.

Hvis 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 søkefeltet 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 eget språk med en egen syntaks, som er forklart i dette dokumentet.

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.

Filterdelen av spørringen vil bestemme utvalget av data. Her vises også et utvalg og så er 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.

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": Dette er en liten tabell med variabler som starter med "[" og avslutter med "]". For tabeller som ikke inneholder aggregerte data, skal gruppering være []. Gruppering inneholder variabler. Her kan vi bestemme nivået på aggregeringen. Forslaget starter med to nivå. Nivåene kan bruker bestemme og bestemmer dybden på tallene.
- sortBy: Her gjelder samme regler som i SQL.

GroupBy:

Eksempel groupBy: ["Institusjonskode", "Avdelingskode", "Årstall"]. Samme regler som i 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

Eksemplene 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 og greaterthan. Disse kan kombineres med ulike verdier. Disse er i anførselstegn("<nnnn>"), adskilt med komma.

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 av variabelen. Verdier er alltid bare **ett** tall. Ikke flere.

```
{
  "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. Tilsvarende 'between' i sql.

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

greaterThan: Typisk årstall yngre enn et gitt år.

```
{
  "variabel": "Årstall",
  "selection": {
    "filter": "greaterthan",
    "values": [
      "2014"
    ]
  }
}
```

Dette eksempelet gir oss filter på høstsemester.

For å unngå store datamengder, lønner det seg å sette opp spørringen med 'item'. 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.

Eksempel på spørring med mange valg:

```
{"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:

```

"filter": [
  {
    "variabel": "Institusjonskode",

```

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

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. 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.
"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".
Deretter kommer data{}

Det er foreløpig lagt inn en grense på 500000 poster pr. spørring. Søket bør deles når det er store datamengder.

Statuskode

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

Status 200 indikerer at request er ok.

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 og skyldes gjerne feil ved JSON inn eller innholdet i denne.

Returkoder i 500 segmentet indikerer et server-side problem, slik at forespørselen ikke kan utføres. Du får det også 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 /hentCSVTabellData eller /hentJSONTabellData. For det siste valget er det mulig å få ut en statuslinje.

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

Hent token:

```
curl -X POST 'https://sso.nsd.no/oauth/token?grant_type=client_credentials&client_id=<klient-id>&client_secret=<secret-id> &scope=%20users:read' > ./token.txt
```

Klient-id og secret-id må institusjonen selv administrere via NSD sitt brukersystem når dette er klart. Foreløpig må institusjonen gi beskjed til NSD om å legge inn klientmaskinen.

Eksempel på cUrl spørring av rapport 112.:

```
export JWT=<sett inn token som er hentet her>
```

```
curl -o ./Result/112-result.dat -i -d "@./queryBody/112-querybody.json" -H "Content-Type: application/json;charset=UTF-8" --header "Authorization: Bearer $JWT" -v -X POST https://api.nsd.no/dbhapitjener/Tabeller/hentCSVTabellData 2>&1
```

Her blir query hentet fra ./queryBody/112-querybody.json og data lagt i /Resultat/112-resultat.dat Etter først å ha fått et gyldig token (pålogging) fra NSD.

Spørring uten token/pålogging med CSV returformat.:

```
:set fileformat=unix
```

```
#!/bin/bash
```

```
echo "Starter uthenting av fil 112."
```

```
export JWT=
```

```
curl -o ./Result/get1-csv-result.dat -i -d "@./queryBody/get1-csv-query.json" -H "Content-Type: application/json;charset=UTF-8" --header "Authorization: Bearer $JWT" -v -X POST https://api.nsd.no/dbhapitjener/Tabeller/hentCSVTabellData>&1
```

eller med Json retur:

```
:set fileformat=unix
```

```
#!/bin/bash
```

```
echo "Starter uthenting av fil 112."
```

```
export JWT=
```

```
curl -o ./Result/get1-json-result.dat -i -d "@./queryBody/get1-json-query.json" -H "Content-Type: application/json;charset=UTF-8" --header "Authorization: Bearer $JWT" -v -X POST https://api.nsd.no/dbhapitjener/Tabeller/hentJSONTabellData>&1
```

-i gir statusinformasjon og kan tas bort etter ønske.

Eksempel på AJAX query fra klient

(Representert ved javascript biblioteket AXIOS).

stitalapi = <sti til api>

jsonquery = 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: jsonquery
},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.

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. På Linux må det kjøres 'fromdos' kommando, eller tilsvarende, for å konvertere tegnsett.

```

{"tabell_id":132,"api_versjon":1,"statuslinje":"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 et program.

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++, textpad eller lignende.

Eksempel på spørring

Detaljert 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"
        ]
      }
    }
  ]
}
```

```
}  
}  
]}
```

Utradisjonell 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"  
      ]  
    }  
  }  
]}
```

Et steg videre

Eksempel på spørring:

Tabell: 135-Utenlandske studenter

Beskrivelse: <http://dbh.nsd.uib.no/dokumentasjon/tabell.action?tabellId=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)

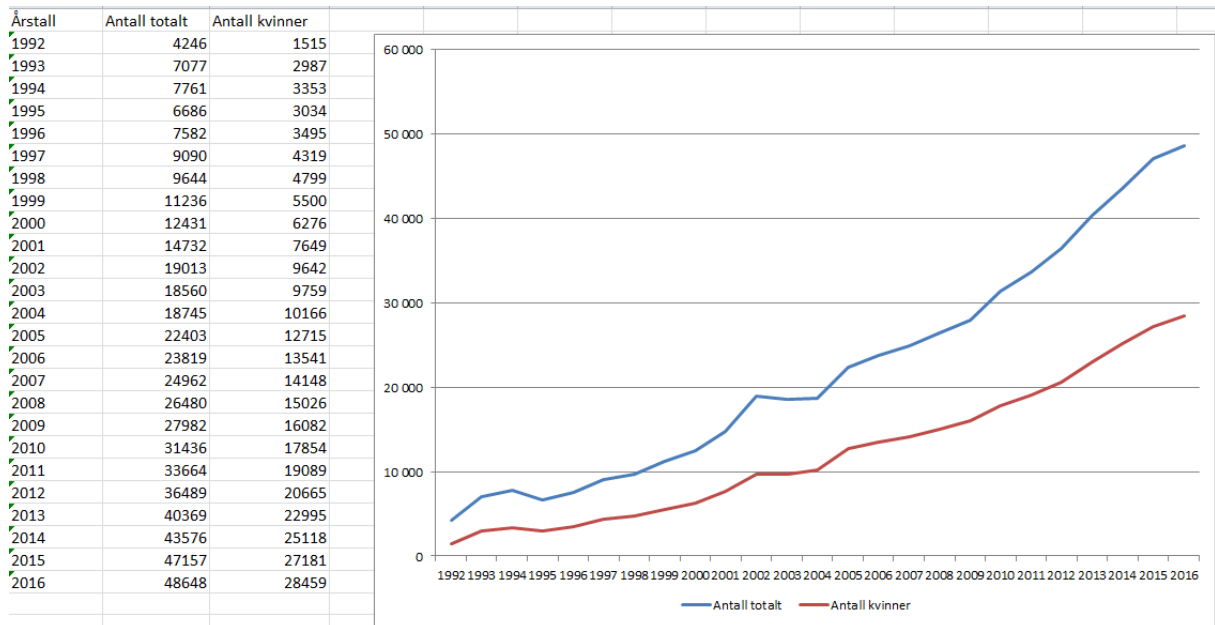
Inngang: <https://api.nsd.no/dbhaptjener/Tabeller/>
Velg mellom: /hentCSVTabellData eller /hentJSONTabellData

Spørring:

```
{
  "tabell_id":135,"api_versjon":1,"statuslinje":"N","decimal_separator":".",
  "groupBy":["Institusjonskode","Avdelingskode","Årstall"],
  "sortBy":["Institusjonskode","Avdelingskode"],
  "filter":[
    {
      "variabel":"Årstall",
      "selection":{
        "filter":"all",
        "values": [
          "2021"
        ]
      }
    }
  ]
}
```

Test spørring - CSV Test spørring - JSON Last ned

Disse dataene kan vi legge over i Excel.



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

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.

Ved eventuelle feil og mangler kan(og bør) DBH kontaktes.

Kilder til inspirasjon for arkitektur:

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

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

Tips og triks:

- MS Explorer. I noen versjoner virker ikke 'fra JSON til Excel'. Bruke Spørring – CSV og importer .txt fil i Excel.
- Store datamengder i klient: Nettleser kan gå tom for minne. Tøm minnet og eventuelt dele opp søket. Du vil ikke få samme problemene om du bruker cUrl script.
- 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) må man velge 'item' og skrive inn institusjonsnr i klient. "<Skriv inn institusjonsnr>"
-
- Du sparer ca. 70 % filstørrelse om du henter i filformat CSV. Dette filformatet anbefales.
- 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 hente data som CSV. Laster ned fil og klipper på nedlastingsfilen. Denne skal da komme opp i Excel. Kan bruke Excel til å søke i filen.