



# StatBank PxWebApi user guide

---

## Content

1. Introduction .....	2
2. Starting the API .....	2
API console.....	2
Search table in console .....	2
Console parts.....	3
3. JSON query .....	4
Output response formats .....	4
4. Post query .....	5
5. Navigate API using URL .....	5
a. Table number .....	5
b. Hierarchy.....	5
Search tables with URL - ?query .....	6
6. Table metadata .....	7
7. Difference between API:Readymade datasets and API:Create your own datasets (PxWebApi).....	7
CSV files are different.....	8
CSV in API: Create your own datasets.....	8
8. Resource Pages .....	8
How to use StatBank.....	8
JSON and JSON-Stat .....	8
Example code.....	9
Curl .....	9

## 1. Introduction

Statistics Norway now offer an API for queries towards all StatBank's 6000 tables. You can grab the entire table or parts of a table. This is an addition to the former API with readymade data sets on <https://data.ssb.no/api/v0/dataset/?lang=en>.

To use the API, it is an advantage to be familiar with creating tables in StatBank and the http protocol. This API uses http POST, not http GET. Queries can be formulated in JSON (JavaScript Object Notation).

URL is composed as follows: `http://data.ssb.no/api-name/api-version/language/`

There are two ways to find tables in the new API:

1. Via console - <https://data.ssb.no/api/v0/en/console/>
2. Via URL - <https://data.ssb.no/api/v0/en/table/>
3. Via search - <https://data.ssb.no/api/v0/en/table/?query=<argument>>

We use the license Creative Commons Attribution 4.0 International (CC BY 4.0).

We welcome your feedback. Send message about errors and shortcomings as well as questions and wishes to: [statistikbanken@ssb.no](mailto:statistikbanken@ssb.no)

## 2. Starting the API

### API console

A console assists the user to generate and edit the code to be used in the API. It is intended to test queries towards the API and not in production. See also a short [video introduction to the console](#).

### Search table in console

In the Table pane: Enter table number or search text. Autocomplete turns on after 3 characters. Then click the selected table.

Table	sitc
	<ul style="list-style-type: none"> <li>08806: External trade in goods, by commodity group (one- and two-digit SITC) and country/trade region/continent 1988M01 - 2016M03</li> <li>08809: External trade in goods, by commodity group (one- and two-digit SITC) and country/trade/region/continent 1988 - 2015</li> </ul>
Variables	<ul style="list-style-type: none"> <li>08818: External trade in goods, by commodity group (three digit SITC) (NOK 1 000) 1988M01 - 2016M03</li> <li>08819: External trade in goods, by commodity group (three digit SITC) (NOK 1 000) 1988 - 2015</li> </ul>
URL	08817: Mainland exports, by county of production and commodity group (SITC based) (NOK million) (C) 1998M01 - 2016M03

select from list

Once you've selected a table, the console will generate an example query in JSON with the first three values for each variable.

# API console

The screenshot shows the API console interface with the following components and callouts:

- 1** Table: 08801: External trade in goods, by commodity number (HS) and country 1988 - 2017
- 2** Variables:
  - commodity number | Varekoder
  - imports/exports | ImpEks
  - country | Land
  - contents | ContentsCode
  - year | Tid
- 3** [Show all metadata in new tab.](#)
- 4** URL: https://data.ssb.no/api/v0/en/table/08801
- 5** Query:
 

```
{
  "query": [
    {
      "code": "Varekoder",
      "selection": {
        "filter": "item",
        "values": [
          "00000011",
          "00000031",
          "00000032",
          "99319900",
          "99999999"
        ]
      }
    }
  ]
}
```
- 6** Run
- 7** Status code: 200
- 8** Result:
 

```
{
  "ImpEks": {
    "label": "imports/exports",
    "category": {
      "index": {
        "1": 0,
        "2": 1
      },
      "label": {
        "1": "Imports",
        "2": "Exports"
      }
    }
  },
  "Land": {
    "label": "country",

```
- 9** Save API query

## Console parts

1. **Search** table by table number or text in title
2. Metadata overview for the table, shows variables
3. Link to all metadata - opens in new tab (see: 2)
4. POST URL. The url you post your query
5. Query. Window where you can edit the generated query in JSON (see: 4)
6. Run - POST query
7. Http status code - 200 (green) if OK
8. Result. The format JSON-stat, version 2 is default. The content shows in the result window when JSON-stat is selected. Downloaded file opens in a separate application, such as Excel.
9. Save or open API query ( .json-file )

### 3. JSON query

All queries start with the object: {query: {...}}. The query is posted to the table URL.



Filters supported:

- **Item** - single values in a list (item is default in console)
- **All** - wildcard \*, e.g. only "\*" returns all values. '2018\*' gives all values starting with 2018



- **Top** - output the number of the latest values



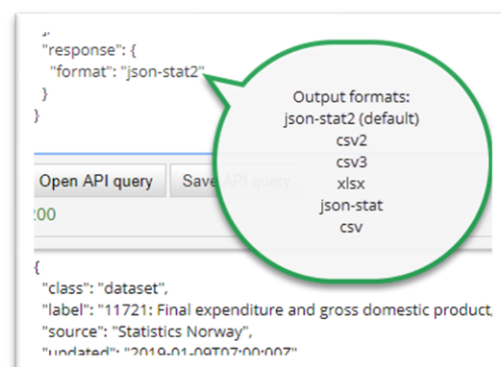
An important characteristic of the filters is to make the query smaller and more transparent. With the filters All and Top, it is possible to construct a rolling query. Then the same query can be used to retrieve new figures.

#### Output response formats

The API can from 2019 return results in 6 formats<sup>1</sup>:

- JSON-stat, version 2 (default)
- CSV2 - pivot-friendly format
- CSV3 - as csv2 with codes instead of text
- XLSX
- JSON-stat v. 1.2
- CSV

Decimal sign is. (period).



<sup>1</sup> It is also possible to get the formats PX (PC-Axis), Statistics Sweden's PX-JSON format and SDMX. These are not supported, but available in the API, as "px", "json" and "sdmx".

If response part is missing in query, the API returns JSON-stat.

## 4. Post query

The API uses HTTP POST, and does not support http GET. In the console you post your query by clicking "Run".

Status codes and error messages:

**200** - OK

**400** – “Bad request” - errors in syntax of the query.

**403** - blocking when querying for large data sets. The API limit is 800,000 cells.

**429** - Too many queries within a minute. The API limit is 30 queries within 60 seconds. Run large queries in sequence. Get the result of the first, before you run the next.

**503** - Time-out after 60 seconds. It may turn on, when extracting large XLSX datasets. (We do not recommend that).

## 5. Navigate API using URL

### a. Table number

If you know the table number in StatBank for the current table, you can use the URL on the form:

[https://data.ssb.no/api/v0/en/table/\(5-digit table number\)](https://data.ssb.no/api/v0/en/table/(5-digit table number))

Example: <https://data.ssb.no/api/v0/en/table/11000>

GET shows table's metadata as JSON. Use the same URL to post the query.

### b. Hierarchy

<https://data.ssb.no/api/v0/en/table/>

retrieves main topics in JSON format from topic hierarchy.

```
[{"id":"al","type":"1","text":"Labour market and earnings"},
{"id":"bf","type":"1","text":"Banking and financial markets"},
{"id":"be","type":"1","text":"Population"},
{"id":"bb","type":"1","text":"Construction, housing and property"},
{"id":"ei","type":"1","text":"Energy and manufacturing"},
{"id":"he","type":"1","text":"Health"},
{"id":"if","type":"1","text":"Income and consumption"},
{"id":"in","type":"1","text":"Immigration and immigrants"},
{"id":"js","type":"1","text":"Agriculture, forestry, hunting and fishing"},
{"id":"kf","type":"1","text":"Culture and recreation"},
{"id":"nk","type":"1","text":"National accounts and business cycles"},
{"id":"nm","type":"1","text":"Nature and the environment"},
{"id":"os","type":"1","text":"Public sector"},
{"id":"pp","type":"1","text":"Prices and price indices"},
{"id":"sk","type":"1","text":"Social conditions, welfare and crime"},
{"id":"sv","type":"1","text":"Svalbard"},
{"id":"ti","type":"1","text":"Technology and innovation"},
{"id":"tr","type":"1","text":"Transport and tourism"},
{"id":"ud","type":"1","text":"Education"},
{"id":"ut","type":"1","text":"External economy"},
{"id":"va","type":"1","text":"Elections"},
{"id":"vt","type":"1","text":"Wholesale and retail trade and service activities"},
{"id":"vf","type":"1","text":"Establishments, enterprises and accounts"}]
```

In general the structure is: *topic - subtopic - statistics - table*. Statistics with many tables may also have one or two levels between statistics and table.

E.g. <https://data.ssb.no/api/v0/en/table/be> list all subtopics in the topic population.

Note that statistics can be placed under several topics and subtopics. All tables under subject id's "in" and "sv" will also be placed elsewhere. POST towards all such addresses will work.

Url 's in the form table number: [https://data.ssb.no/api/v0/en/table/\(5-digit table number\)](https://data.ssb.no/api/v0/en/table/(5-digit table number))

is more stable over time, than the url of the form:

[https://data.ssb.no/api/v0/en/table/\(topic\)/\(subtopic\)/\(statistics\)/\(tablename\)](https://data.ssb.no/api/v0/en/table/(topic)/(subtopic)/(statistics)/(tablename)).

This is because the tables in some cases can be moved in the subject structure.

"type":

- "l" - sublevel
- "t" - table

Example from <https://data.ssb.no/api/v0/en/table/be/be02/folkfram>:

```
{
  "id": "Framskr2016T6",
  "type": "t",
  "text": "11172: Projected life expectancy, for men, women and both sexes combined, in 3
  variants 2015 - 2100",
  "updated": "2016-06-21T10:00:00"
}
```

At this level means the different tags:

- "id": tablename
- "type": t = Table
- "text": Table title (5-digit table number + description of contents + timespan)

## Search tables with URL - ?query

? query - search variables and titles. The search is case-insensitive. Space gives AND. The filter "title" narrows your search to the title field. You can also narrow by using field name plus colon, eg. *title:* . The search engine is Lucene v. 2.9.4 (win).

Examples:

- Search for "cake mix" in all tables

<https://data.ssb.no/api/v0/en/table/?query=cake mix>

- Use table id to find table number:

<https://data.ssb.no/api/v0/en/table/?query=folkframt6>

- Search all tables within topic population for the value (municipality) "Hattfjelldal» using mask \*

[https://data.ssb.no/api/v0/en/table/be/?query=hattfjell\\*](https://data.ssb.no/api/v0/en/table/be/?query=hattfjell*)

- Search "item number" and "HS" less than 5 words apart

[https://data.ssb.no/api/v0/en/table/?query=commodity number hs "~ 5](https://data.ssb.no/api/v0/en/table/?query=commodity number hs )

- Search urban district tables under area population

[https://data.ssb.no/api/v0/en/table/be/?query=title:\(UD\)](https://data.ssb.no/api/v0/en/table/be/?query=title:(UD))

- List all tables in Statbank.

[https://data.ssb.no/api/v0/en/table/?query=\\*&filter=\\*](https://data.ssb.no/api/v0/en/table/?query=*&filter=*)

- Search tables published a specific date.

[https://data.ssb.no/api/v0/en/table/?query=20180507\\*&filter=published](https://data.ssb.no/api/v0/en/table/?query=20180507*&filter=published)

Mark that the characters "(), and spaces will be escaped by respectively. % 22,% 28,% 29 and% 20

## 6. Table metadata

Http GET to the URL you post to, will return main the table's metadata as JSON.

The metadata consists of a title "title" and a list of variables for the table. Variable objects have four attributes :

- Code (code) - mandatory
- Name (text) - mandatory
- Elimination (elimination)
- Time (time)

Example from <https://data.ssb.no/api/v0/en/table/11172>:

```
[
  {
    title: "Projected life expectancy, by sex, age, variants and year",
    variables: [
      {
        code: "Kjonn",
        text: "sex",
        values: [
          "0",
          "1",
          "2"
        ],
        valueTexts: [
          "Both sexes",
          "Males",
          "Females"
        ],
        elimination: true
      },
      {
        code: "Alder",
        text: "age",
        values: [
          "000",
          "001",
          "002",
          "003"
        ]
      }
    ]
  }
]
```

Variable objects contain two lists, a list of codes and with presentation texts.

It is possible to remove a variable from the query.

If elimination is true, a sum value will be returned. If not, all single values will be returned.

Only one variable can have property "time" set to true. If the "time" variable is eliminated, all values are listed.

Groupings and aggregations are missing in metadata. There are two more filters: Vs and Agg.

In StatBank below the finished table in [API query for this table](#), it is possible to see these filters for your selection. Groupings can be e.g. within "Municipalities entire code list" the current municipality division: "Municipalities 2019" and aggregations can eg. be "5-year age groups". Th filter Vs is redundant and the expression can be replaced by "filter" : "item"

## 7. Difference between API:Readymade datasets and API:Create your own datasets (PxWebApi)

In the PxWebApi users formulate their own queries towards 6000 different tables, while the API from 2013 serves readymade data sets. Towards these 250 readymade datasets you can use http GET. If you find your dataset on <https://data.ssb.no/api/v0/dataset/?lang=en> is an easy solution with stable URLs. It is possible to contact [statistikkbanken@ssb.no](mailto:statistikkbanken@ssb.no) to request new datasets.

## CSV files are different

### CSV in API with readymade datasets:

- One variable per line
- decimal separator: , (comma) in Norwegian and . (dot) in English
- field delimiter ; (semicolon)

### CSV in API: Create your own datasets

- field delimiter comma,
- decimal separator . (dot)
- 3 different CSV formats. CSV2 is quite similar to the CSV in readymade datasets.

For a flexible treatment of CSV we recommend JSON-stat as output, and then the [JSON-stat Command Line Conversion Tools](#) (requires node.js)

## 8. Resource Pages

### How to use StatBank

Before using the API users should be used to the StatBank:

<https://www.ssb.no/en/informasjon/om-statistikkbanken/how-to-use-statbank-norway>

Tables in StatBank can be accessed by the address <https://www.ssb.no/en/table/5-digit-table> number

E.g.: <https://www.ssb.no/en/table/05000>

We basically use the same API code as Statics Sweden (SCB), plus our own developed console. See [https://www.scb.se/api\\_en/](https://www.scb.se/api_en/) and SCB's [API guide](#).

.	<b>Not applicable</b> Figures do not exist at this time, because the category was not in use when the figures were collected.
..	<b>Not available</b> Figures have not been entered into our databases and are therefore unknown.
:	<b>Confidential</b> Figures are not published to avoid identifying persons or companies.

### JSON and JSON-Stat

Although API has six output formats, we recommend using JSON-stat version 2. This version is also the output from Eurostat, and all known libraries support this. See <https://json-stat.org/> and the [JSON-stat Toolkit](#).

There are available JSON-stat libraries for:

- Javascript
- Python,
- R
- Java
- Julia



**Example code**

Code examples for Statistics Norway's API using different libraries for JSON-Stat, R, Python, Highcharts with jquery, Excel etc., see: [PxWebApi Code examples](#)

In the browser, users should have a JSON viewer, In Firefox this is installed by default.

For Chrome(ium) try this extension: <https://jsonview.com>

**http POST**

As alternative to the console we can recommend Postman <https://www.getpostman.com>

**Curl**

In production, a tool like Curl may be suitable for posting queries. See <https://curl.haxx.se/>.

If you save a query from the console to file, query.json. @ reads from file, the option -d or --data is used for http POST and -o to save the output, e.g.

```
> curl -d @query.json -X POST https://data.ssb.no/api/v0/en/table/08801 -o resultfile
```