

# Getfame

names  
series  
expressions

FAME Json api 2024 / 2025

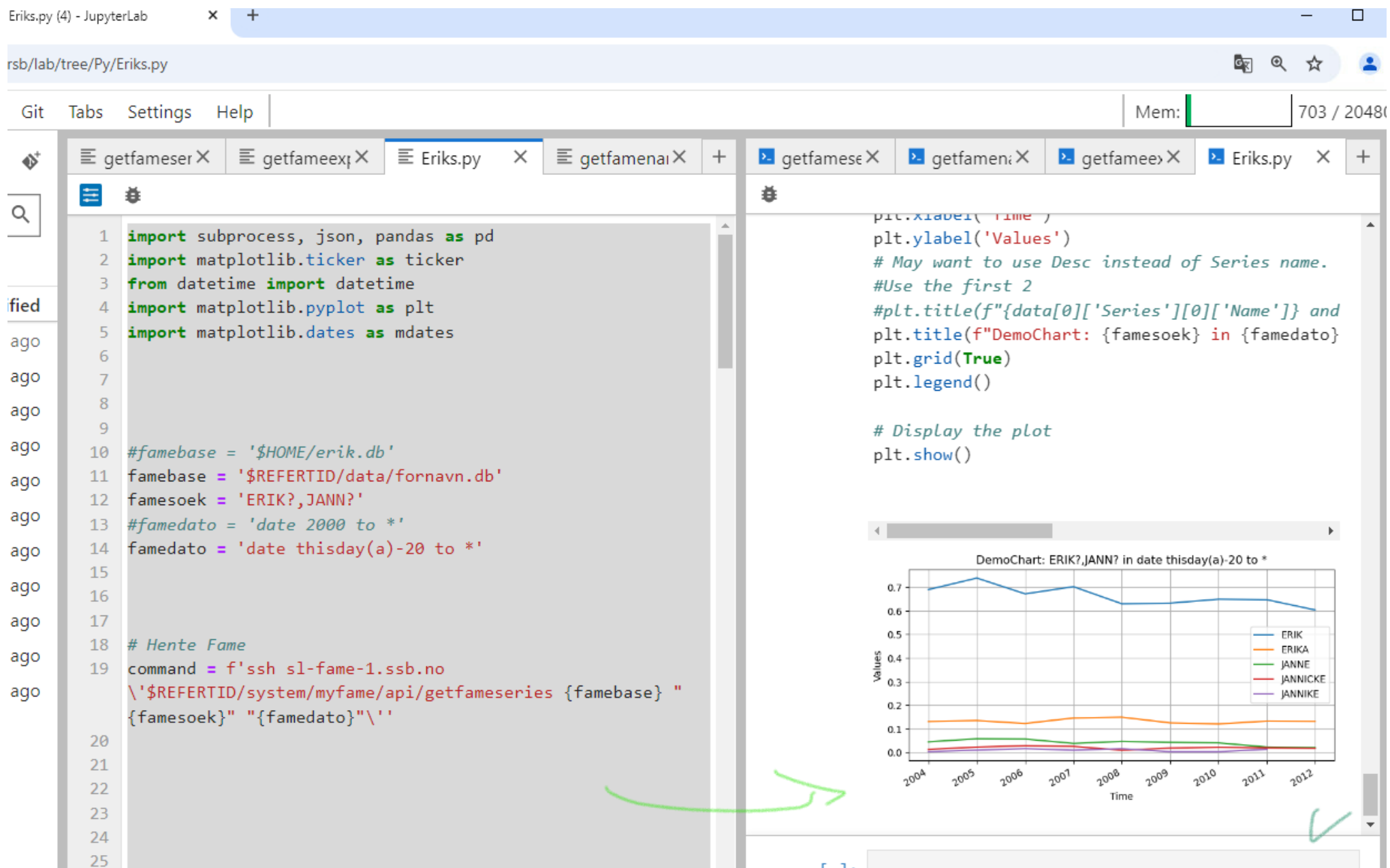
Erik

2025 Supports series w identical series names in different database, formulas can aggregate from several open database

# From Xterm (start here to understand)

```
sl-fame-1
sl-fame-1:/ssb/bruker/refertid/system/myfame/api> /ssb/bruker/refertid/system/myfame/api/getfameexpr /ssb/bruker/refertid/data/kpi_publ.db "pct(total.ipr)" "date 2023 to *"
[{"Getfamejsonapi": "ErikSSB",
"Version": "Kampala20241001",
"Executed": "2024-10-09T12:21:25",
"Famever": "11.53",
"Database": "/ssb/bruker/refertid/data/kpi_publ.db",
"Result": "$HOME/.GetFAME/getfameexpr.json",
"Series": [
{"Name": "PCT(TOTAL.IPR)",
"Desc": "pct(total.ipr)",
"Daterange": "2023 TO *",
"Frequency": "MONTHLY",
"Observations": [
{"Date": "2023-01-01", "Value": 0.1588562, "Epo": [1672531200000, 0.1588562]},
{"Date": "2023-02-01", "Value": 0.3965107, "Epo": [1675209600000, 0.3965107]},
{"Date": "2023-03-01", "Value": 0.7898894, "Epo": [1677628800000, 0.7898894]},
{"Date": "2023-04-01", "Value": 1.097179, "Epo": [1680307200000, 1.097179]},
{"Date": "2023-05-01", "Value": 0.4651163, "Epo": [1682899200000, 0.4651163]},
{"Date": "2023-06-01", "Value": 0.617284, "Epo": [1685577600000, 0.617284]},
{"Date": "2023-07-01", "Value": 0.3834356, "Epo": [1688169600000, 0.3834356]},
{"Date": "2023-08-01", "Value": -0.7639419, "Epo": [1690848000000, -0.7639419]},
{"Date": "2023-09-01", "Value": -0.07698229, "Epo": [1693526400000, -0.07698229]},
{"Date": "2023-10-01", "Value": 1.001541, "Epo": [1696118400000, 1.001541]},
{"Date": "2023-11-01", "Value": 0.5339436, "Epo": [1698796800000, 0.5339436]},
{"Date": "2023-12-01", "Value": 0.07587253, "Epo": [1701388800000, 0.07587253]},
{"Date": "2024-01-01", "Value": 0.07581501, "Epo": [1704067200000, 0.07581501]},
{"Date": "2024-02-01", "Value": 0.2272727, "Epo": [1706745600000, 0.2272727]},
{"Date": "2024-03-01", "Value": 0.2267574, "Epo": [1709251200000, 0.2267574]},
{"Date": "2024-04-01", "Value": 0.8295626, "Epo": [1711929600000, 0.8295626]},
{"Date": "2024-05-01", "Value": -0.1495886, "Epo": [1714521600000, -0.1495886]},
{"Date": "2024-06-01", "Value": 0.2247191, "Epo": [1717200000000, 0.2247191]},
{"Date": "2024-07-01", "Value": 0.5231689, "Epo": [1719792000000, 0.5231689]},
{"Date": "2024-08-01", "Value": -0.8921933, "Epo": [1722470400000, -0.8921933]}
] } 1,
"Elapsed_time_seconds": 0.257
} 1
sl-fame-1:/ssb/bruker/refertid/system/myfame/api>
```

From  
jupyter  
using  
py:



# 1. getfamenames `$REFERTID/system/myfame/api/getfamenames`

- List FAME series & formulas with **metadata**, from fame database(s) given a list of series/wildcards:

```
$REFERTID/system/myfame/api/getfamenames "$REFERTID/data/cpi.db, erik.db" "K011?.ipr, k09?.ipr"
```

"Series":[

```
{"Name":"CPI'K011.IPR","Class":"SERIES","Observed":"AVERAGED","Desc": "Food","Created":"2017-01-18T18:28:28","Updated":"2024-09-10T08:54:45"},
```

```
{"Name":"CPI'K0111.IPR","Class":"SERIES","Observed":"AVERAGED","Desc":"Bread pris","Created":"2017-01-18T18:28:28","Updated":"2024-09-10T08:54:45"},
```

```
{"Name":"CPI'K01111.IPR","Class":"SERIES","Observed":"AVERAGED","Desc":"Rice_index price","Created":"2017-01-18T18:28:28","Updated":"2024-09-10T08:54:44"},
```

```
{"Name":"CPI'K01111_11111.IPR","Class":"SERIES","Observed":"AVERAGED","Desc":"Ris_indeks pris","Created":"2017-01-18T18:28:28","Updated":"2024-09-10T08:54:44"},
```

```
{"Name":"CPI'K01112.IPR","Class":"SERIES","Observed":"AVERAGED","Desc":"Mel og andre kornprodukter_indeks pris","Created":"2017-01-18T19:24:41","Updated":"2024-09-10T08:54:44"},
```

```
{"Name":"CPI'K01112_11121.IPR","Class":"SERIES","Observed":"AVERAGED","Desc":"Mel_indeks pris","Created":"2017-01-18T18:28:28","Updated":"2024-09-10T08:54:44"}
```

.

## 2. getfameseries `$REFERTID/system/myfame/api/getfameseries`

- **Data-observations, from a fame database given a list of wildcards, and optional daterange:**

**getfameseries `"/ssb/bruker/refertid/data/kpi_publ.db, $HOME/kpi.db" "K01199?.lpr, K8?" "freq m; date jul24 to aug24; deci 1"`**

```
"Series": [  
  {"Name": "K01199.IPR",  
    "Desc": "Andre matvarer ikke ellers nevnt_indeks pris",  
    "Daterange": "JUL24 TO AUG24",  
    "Frequency": "MONTHLY",  
    "Observations": [  
      {"Date": "2024-07-01", "Value": 142.4, "Epo": [1719792000000, 142.4]},  
      {"Date": "2024-08-01", "Value": 135.3, "Epo": [1722470400000, 135.3]}  
    ] } ,  
  {"Name": "K01199_11991.IPR",  
    "Desc": "Supper og kraft_indeks pris",  
    "Daterange": "JUL24 TO AUG24",  
    "Frequency": "MONTHLY",  
    "Observations": [  
      {"Date": "2024-07-01", "Value": 151.6, "Epo": [1719792000000, 151.6]},  
      {"Date": "2024-08-01", "Value": 143.3, "Epo": [1722470400000, 143.3]}  
    ] } ,  
]
```

### 3. getfameexpr `$REFERTID/system/myfame/api/getfameexpr`

advanced mode

- Data-observations, from FAME database(s) given a fame-**expression**:

```
getfameexpr " $REFERTID/data/fornavn.db " "mave(ERIK,2)" "date 2000 to 2010"
```

```
getfameexpr " $REFERTID/data/fornavn.db " "Lsum(ERIK,EIRIK)" "date 2000 to *"
```

```
getfameexpr " $REFERTID/data/kpi_publ.db,mycpi.db " "convert(total.ipr,annual,constant)" "date *; deci 1 "
```

```
getfameexpr " $REFERTID/data/kpi_publ.db,mycpi.db " "PCT(mycpi'K09.IPR)" "date 2025; deci 1"
```

Produces same json outp..ut as getfameseries

# Samples ( shows help info, when no arguments passed)

1. \$REFERTID/system/myfame/api/getfamenames
2. \$REFERTID/system/myfame/api/getfameseries
3. \$REFERTID/system/myfame/api/getfameexpr

For **jupyterlab** sample, see:

- <https://github.com/statisticsnorway/getfame-json-api/blob/main/sample.py>

# More getfameseries samples:

\$REFERTID/system/myfame/api/getfameseries /ssb/bruker/refertid/data/kpi\_publ.db "total.ipr"

\$REFERTID/system/myfame/api/getfameseries /ssb/bruker/refertid/data/kpi\_publ.db "total.ipr " "date 2024 "

\$REFERTID/system/myfame/api/getfameseries /ssb/bruker/refertid/data/kpi\_publ.db "total.ipr" "freq m; date thisday(m)-5 to \*"

\$REFERTID/system/myfame/api/getfameseries \$REFERTID/data/fornavn.db "?ERIK,KRISTIN,JIM?}" "date 2010 to 2012 "

\$REFERTID/system/myfame/api/getfameseries /ssb/bruker/refertid/data/fornavn.db "?JAN?" "date 2000 to 2005 "

\$REFERTID/system/myfame/api/getfameseries /ssb/bruker/refertid/data/fornavn.db "JI? " "date 2000 to \*; deci 1 "

\$REFERTID/system/myfame/api/getfameseries "fornavn.db, name.db" "JI? ,MATT?" "date 2000 to \* ; deci 2«

\$REFERTID/system/myfame/api/getfameseries «cpi1.db,cpi2.db,cpi\_form.db" «Total.ipr" "date 2025 ; deci 2"



# More getfameexpr expressions

```
$REFERTID/system/myfame/api/getfameexpr $REFERTID/data/fornavn.db "mave(ERIK,2)" "date 2000 to 2010 "
```

```
$REFERTID/system/myfame/api/getfameexpr $REFERTID/data/fornavn.db "Lsum(ERIK,EIRIK)" "date 2000 to * "
```

```
$REFERTID/system/myfame/api/getfameexpr $REFERTID/data/kpi_publ.db "convert(total.ipr,annual,constant)" "date 2020 to * "
```

--use of custom common **cb** basis to set base-year to 2010 (instead of current 2015)

```
$REFERTID/system/myfame/api/getfameexpr getfameexpr /ssb/bruker/refertid/data/kpi_publ.db "cb(total.ipr,2010)" "date 2010 to 2020"
```

```
$REFERTID/system/myfame/api/getfameexpr $REFERTID/data/kpi_publ.db "total.ipr" "freq m; date jan20 to feb20;deci 0 "
```

```
$REFERTID/system/myfame/api/getfameexpr «cpi1.db,cpi2.db,cpi_form.db" "cpi1'Total.ipr" "date 2025 ; deci 2"
```

```
$REFERTID/system/myfame/api/getfameexpr $REFERTID/data/fornavn.db "ERIK+EIRIK" "date 2000 to 2010;deci 1"
```

# Using R from Jupyterlab

