

Brukerveiledning "datamodeller"-API

Innholdsfortegnelse

- [Brukerveiledning "datamodeller"-API](#)
 - [Innholdsfortegnelse](#)
 - [Innledning](#)
 - [Datakilder](#)
 - [Datamodell](#)
 - [Turnusdata](#)
 - [APIer](#)
 - [Tilgang og autentisering](#)
 - [Deltalast](#)
 - [Begrensninger](#)
 - [Eksempel på bruk](#)
 - [Uthenting av data](#)
 - [Python med Requests](#)
 - [PowerShell med RestMethod](#)
 - [cURL](#)
 - [Ekspandere data til daglig nivå](#)
 - [Rekursiv datoekspansjon](#)
 - [Ekspansjon med kalendertabell](#)
 - [Koble mot andre objekter](#)

Innledning

Denne brukerveiledningen er ment å gi en oversikt over APIene som er laget for å hente ut styringsdata og eksempel på bruk av disse.

APIene gir tilgang til modellert data som forenkler det videre arbeidet med dataforedling. Samtidig forsøker vi å legge minimalt med føringer på hvordan foredlingsarbeidet skal gjøres, slik at det kan tilpasses ulike behov. APIene er ment å konsumeres via datavarehus og krever ytterligere sammenstilling og behandling før dataen kan presenteres i f. eks. rapporteringsverktøy.

Datakilder

Den modellerte dataen kommer fra DFØ sitt SAP kjernesystem. I tillegg er det tilgjengeliggjort rådata fra turnussystemet GAT.

Datamodell

Datamodellen er delt inn i tre hovedområder som har hvert sitt API-endepunkt:

- ansatteobjekter
- orgobjekter
- stillingobjekter

ansatteobjekter inneholder informasjon om de ansatte og ligger tett opp mot de ulike infotypene man finner i SAP. Dette området omfatter også tid-, reise- og lønnsdata.

stillingobjekter inneholder informasjon om stillingers plassering, samt yrkeskoder og stillingsgrupper.

orgobjekter består av organisasjonshierarkiet som inkluderer kostnadssted.

De tre områdene kan kobles sammen via objektene `ansattPlassering` og `stillingPlassering` som har gjensidige referanser, i tillegg til referanser mot organisasjonshierarkiet. Objektene kan også kobles videre mot data fra eksisterende APler for å legge til mer informasjon ved behov.

Turnusdata

Et utvalg av turnusdata er tilgjengeliggjort via API-endepunktet **turnustabeller**. Her finner man informasjon om blant annet vakter, overtid, fravær, banker og saldoer med mer. Dette havner på siden av datamodellen siden det er hentet fra et uavhengig kildesystem. Det finnes allikevel felter, f. eks. ansattnummer (EMP_CODE), som gjør det mulig å sammenstille informasjon fra disse to kildene.

En full oversikt over datamodellen og de ulike objektene finnes på [mirotavle](#).

APler

APlene beskrevet over følger Odata v4 standard. De leveres med tjenestedokument og metadata i xml format som beskriver tjenesten. Samtlige APler er ment for dataauthenting og er begrenset til "get"-kall. Det er støtte for paginering og maksimalt mulig å hente ut 10 000 rader om gangen.

Utfyllende informasjon er tilgjengelig på DFØ sin [API portal](#).

Tilgang og autentisering

APlene til DFØ gjøres tilgjengelig gjennom en felles API manager og Digitaliseringsdirektoratets [Maskinporten](#) brukes for sikker autentisering og tilgangskontroll. API Manager krever autentisering for alle endepunkter og tilgang er kun for system-til-system integrasjoner.

scope	endepunkt
dfo:datamodeller	ansatteobjekter
	stillingobjekter
	orgobjekter
dfo:turnustabeller	turnustabeller

Deltalast

APlene har støtte for deltalast via et continuation-token. Tokenet fungerer som en referanse til siste rad som ble hentet ut ved siste spørring. Det er støtte for å håndtere slettede rader for alle turnustabeller samt ansatteobjektet tidevalueringDag. På andre tabeller eller objekter er det per nå ikke støtte for håndtering av slettede data, så vi anbefaler da kun å bruke deltalast hvis man er sikre på at data i de aktuelle objektene ikke blir slettet.

Detaljer om denne funksjonaliteten og eksempler på bruk av denne er beskrevet på DFØ sin [API portal](#)

Begrensninger

Pga store datavolum har vi vært nødt til å innføre noen restriksjoner.

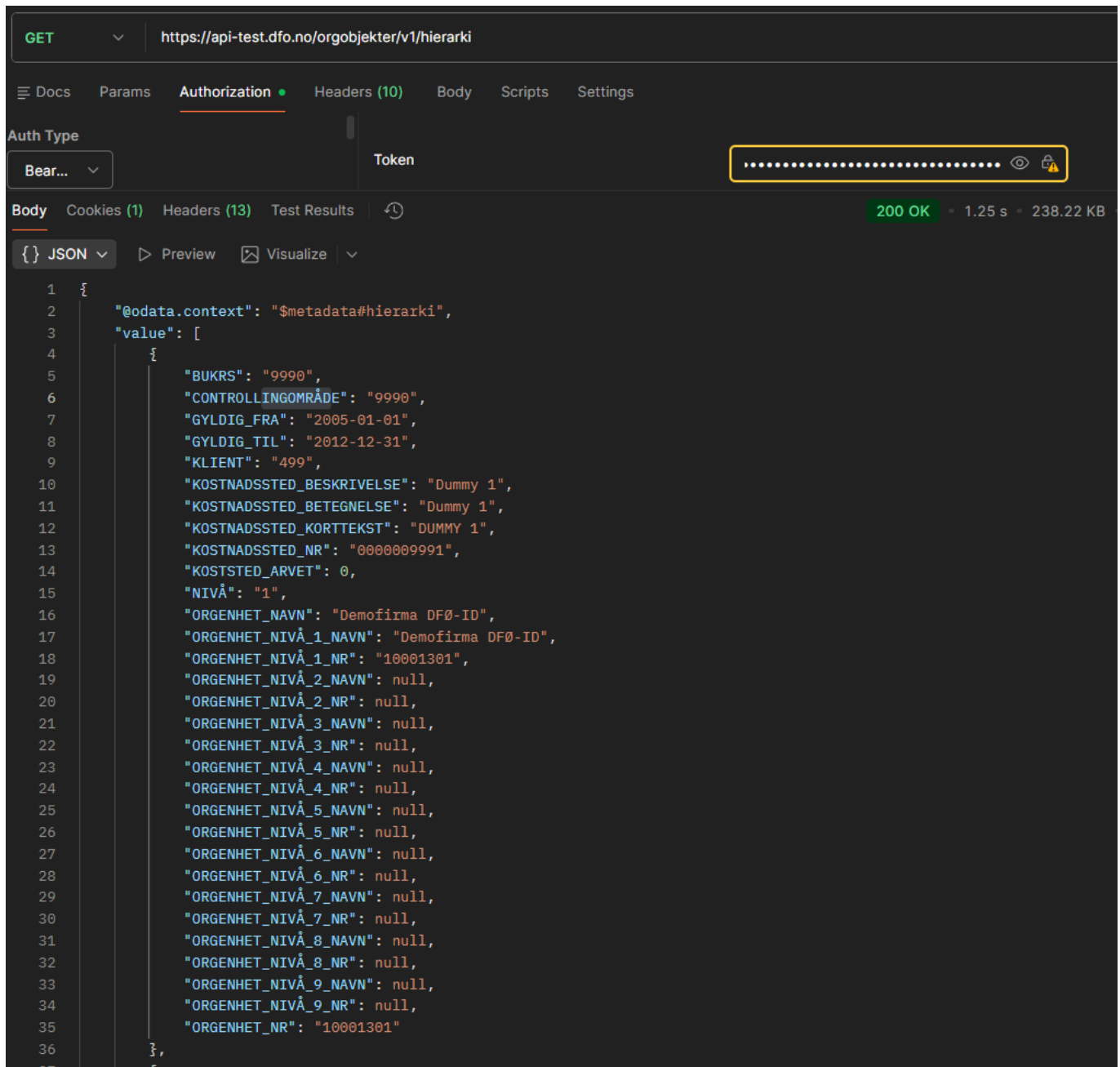
- Tidevaluering på daglig nivå utgir data for forhenværende og inneværende år, frem til dagens dato. Dette følger rekalkuleringsvinduet. Full historikk er tilgjengelig på månedlig nivå.
- Arbeidsplanen på daglig nivå er tilgjengelig 5 år tilbake og 1 år frem i tid. Full historikk er tilgjengelig på månedlig nivå.

Eksempel på bruk

I dette eksempelet henter vi ut organisasjonshierarkiet via APllet orgobjekter og viser hvordan det kan ekspanderes til daglig data. Det gis også noen indikasjoner på hvordan datasettet kan kobles videre mot resten av modellen. Merk at eksemplene benytter fiktive data fra vårt testmiljø. URLer for våre APler i produksjonsmiljøet er tilgjengelig i avsnittet [APler](#), over.

Uthenting av data

Under er et eksempel på hvordan man kan sende en spørring via postman for å hente ut data fra organisasjonshiearkiet. Vi gjør et "get"-kall og benytter autentiseringstypen "Bearer Token". Det forutsettes at man har tilgang til å hente ut jwt-token via maskinporten med nødvendig scope.



Figur 1: Skjerm bilde av respons fra API-kall for å hente ut organisasjonshierarki.

Dette kan tilsvarende utføres maskinelt via ulike kodespråk. Under følger noen eksempler. Merk at `<YOUR_TOKEN>` må erstattes med jwt-tokenet dere henter fra maskinporten.

Python med Requests

```
import requests

url = "https://api-test.dfo.no/orgobjekter/v1/hierarki"

payload = {}
headers = {
    'Authorization': 'Bearer <YOUR_TOKEN>',
}

response = requests.request("GET", url, headers=headers, data=payload)
```

```
print(response.text)
```

PowerShell med RestMethod

```
$headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
$headers.Add("Authorization", "Bearer <YOUR_TOKEN>")

$response = Invoke-RestMethod 'https://api-test.dfo.no/orgobjekter/v1/hierarki' -
Method 'GET' -Headers $headers
$response | ConvertTo-Json
```

cURL

```
curl --location 'https://api-test.dfo.no/orgobjekter/v1/hierarki' \
--header 'Authorization: Bearer <YOUR_TOKEN>'
```

Ekspandere data til daglig nivå

APIet utleverer hovedsaklig poster (rader) med gyldighetsintervaller for å være mest mulig effektivt. Det kan være behov for å utvide dette til daglige data for å forenkle videre koblinger av datasett for å bygge opp analytiske rapporter.

Rekursiv datoekspansjon

Under følger et eksempel på hvordan dette kan gjøres rekursivt i SQLite.

Merk at man må legge på en øvre grense for hvor langt frem i tid man ønsker å generere datasettet. I eksempelet over er grensen satt til **31-12-2025**. Uten denne risikerer man at datasettet genereres helt frem til SAP systemet sin maks grense.

```
WITH RECURSIVE DATE_EXPANSION AS (
  SELECT
    GYLDIG_FRA AS DATO,
    BUKRS,
    CONTROLLINGOMRÅDE,
    GYLDIG_FRA,
    GYLDIG_TIL,
    KLIENT,
    KOSTNADSSTED_BESKRIVELSE,
    KOSTNADSSTED_BETEGNELSE,
    KOSTNADSSTED_KORTTEKST,
    KOSTNADSSTED_NR,
    KOSTSTED_ARVET,
```

```

    NIVÅ,
    ORGENHET_NAVN,
    ORGENHET_NIVÅ_1_NAVN,
    ORGENHET_NIVÅ_1_NR,
    ORGENHET_NIVÅ_2_NAVN,
    ORGENHET_NIVÅ_2_NR,
    ORGENHET_NIVÅ_3_NAVN,
    ORGENHET_NIVÅ_3_NR,
    ORGENHET_NIVÅ_4_NAVN,
    ORGENHET_NIVÅ_4_NR,
    ORGENHET_NIVÅ_5_NAVN,
    ORGENHET_NIVÅ_5_NR,
    ORGENHET_NIVÅ_6_NAVN,
    ORGENHET_NIVÅ_6_NR,
    ORGENHET_NIVÅ_7_NAVN,
    ORGENHET_NIVÅ_7_NR,
    ORGENHET_NIVÅ_8_NAVN,
    ORGENHET_NIVÅ_8_NR,
    ORGENHET_NIVÅ_9_NAVN,
    ORGENHET_NIVÅ_9_NR,
    ORGENHET_NR

FROM
    ORG_HIERARKI
UNION ALL

SELECT
    DATE(DATO, '+1 day') AS DATO,
    BUKRS,
    CONTROLLINGOMRÅDE,
    GYLDIG_FRA,
    GYLDIG_TIL,
    KLIENT,
    KOSTNADSSTED_BESKRIVELSE,
    KOSTNADSSTED_BETEGNELSE,
    KOSTNADSSTED_KORTTEKST,
    KOSTNADSSTED_NR,
    KOSTSTED_ARVET,
    NIVÅ,
    ORGENHET_NAVN,
    ORGENHET_NIVÅ_1_NAVN,
    ORGENHET_NIVÅ_1_NR,
    ORGENHET_NIVÅ_2_NAVN,
    ORGENHET_NIVÅ_2_NR,
    ORGENHET_NIVÅ_3_NAVN,
    ORGENHET_NIVÅ_3_NR,
    ORGENHET_NIVÅ_4_NAVN,
    ORGENHET_NIVÅ_4_NR,
    ORGENHET_NIVÅ_5_NAVN,
    ORGENHET_NIVÅ_5_NR,
    ORGENHET_NIVÅ_6_NAVN,
    ORGENHET_NIVÅ_6_NR,
    ORGENHET_NIVÅ_7_NAVN,
    ORGENHET_NIVÅ_7_NR,
    ORGENHET_NIVÅ_8_NAVN,

```

```
        ORGENHET_NIVÅ_8_NR,  
        ORGENHET_NIVÅ_9_NAVN,  
        ORGENHET_NIVÅ_9_NR,  
        ORGENHET_NR  
FROM  
    DATE_EXPANSION  
WHERE  
    DATO < MIN(GYLDIG_TIL, '2025-12-31')  
)  
SELECT  
    DATO,  
    BUKRS,  
    CONTROLLINGOMRÅDE,  
    GYLDIG_FRA,  
    GYLDIG_TIL,  
    KLIENT,  
    KOSTNADSSTED_BESKRIVELSE,  
    KOSTNADSSTED_BETEGNELSE,  
    KOSTNADSSTED_KORTEKST,  
    KOSTNADSSTED_NR,  
    KOSTSTED_ARVET,  
    NIVÅ,  
    ORGENHET_NAVN,  
    ORGENHET_NIVÅ_1_NAVN,  
    ORGENHET_NIVÅ_1_NR,  
    ORGENHET_NIVÅ_2_NAVN,  
    ORGENHET_NIVÅ_2_NR,  
    ORGENHET_NIVÅ_3_NAVN,  
    ORGENHET_NIVÅ_3_NR,  
    ORGENHET_NIVÅ_4_NAVN,  
    ORGENHET_NIVÅ_4_NR,  
    ORGENHET_NIVÅ_5_NAVN,  
    ORGENHET_NIVÅ_5_NR,  
    ORGENHET_NIVÅ_6_NAVN,  
    ORGENHET_NIVÅ_6_NR,  
    ORGENHET_NIVÅ_7_NAVN,  
    ORGENHET_NIVÅ_7_NR,  
    ORGENHET_NIVÅ_8_NAVN,  
    ORGENHET_NIVÅ_8_NR,  
    ORGENHET_NIVÅ_9_NAVN,  
    ORGENHET_NIVÅ_9_NR,  
    ORGENHET_NR  
FROM  
    DATE_EXPANSION  
ORDER BY  
    KLIENT, ORGENHET_NR, DATO;
```

Ekspansjon med kalendertabell

Et alternativ som gir mer kontroll på utvidelsen er å lage en kalendertabell som man joiner med. Under viser vi et slikt eksempel i SQLite.

Oppretter kalendertabell.

```
CREATE TABLE CALENDAR AS
WITH RECURSIVE calendar_cte(date) AS (
  SELECT DATE('2005-01-01')
  UNION ALL
  SELECT DATE(date, '+1 day')
  FROM calendar_cte
  WHERE date < '2025-12-31'
)
SELECT date FROM calendar_cte
```

Ekspanderer organisasjonshiearkiet til daglig nivå.

```
SELECT
  c.date AS DATO, o.*
FROM ORG_HIERARKI o
JOIN CALENDAR c
  ON c.date BETWEEN o.GYLDIG_FRA AND o.GYLDIG_TIL
WHERE c.date <= '2025-12-31'
```

I begge tilfeller ender vi opp med et resultat tilsvarende skjermbildet under.

DATO	KLIENT	ORGENH...	GYLDIG F...	GYLDIG...	ORGENHET...	BUKRS	NIVÅ	KOSTSTE...	CONTR...
2005-01-01	499	10001301	2005-01-01	2012-12-31	Demofirma DFØ-ID	9990	1	0	9990
2005-01-02	499	10001301	2005-01-01	2012-12-31	Demofirma DFØ-ID	9990	1	0	9990
2005-01-03	499	10001301	2005-01-01	2012-12-31	Demofirma DFØ-ID	9990	1	0	9990
2005-01-04	499	10001301	2005-01-01	2012-12-31	Demofirma DFØ-ID	9990	1	0	9990
2005-01-05	499	10001301	2005-01-01	2012-12-31	Demofirma DFØ-ID	9990	1	0	9990
2005-01-06	499	10001301	2005-01-01	2012-12-31	Demofirma DFØ-ID	9990	1	0	9990
2005-01-07	499	10001301	2005-01-01	2012-12-31	Demofirma DFØ-ID	9990	1	0	9990
2005-01-08	499	10001301	2005-01-01	2012-12-31	Demofirma DFØ-ID	9990	1	0	9990
2005-01-09	499	10001301	2005-01-01	2012-12-31	Demofirma DFØ-ID	9990	1	0	9990
2005-01-10	499	10001301	2005-01-01	2012-12-31	Demofirma DFØ-ID	9990	1	0	9990
2005-01-11	499	10001301	2005-01-01	2012-12-31	Demofirma DFØ-ID	9990	1	0	9990
2005-01-12	499	10001301	2005-01-01	2012-12-31	Demofirma DFØ-ID	9990	1	0	9990
2005-01-13	499	10001301	2005-01-01	2012-12-31	Demofirma DFØ-ID	9990	1	0	9990
2005-01-14	499	10001301	2005-01-01	2012-12-31	Demofirma DFØ-ID	9990	1	0	9990
2005-01-15	499	10001301	2005-01-01	2012-12-31	Demofirma DFØ-ID	9990	1	0	9990
2005-01-16	499	10001301	2005-01-01	2012-12-31	Demofirma DFØ-ID	9990	1	0	9990
2005-01-17	499	10001301	2005-01-01	2012-12-31	Demofirma DFØ-ID	9990	1	0	9990
2005-01-18	499	10001301	2005-01-01	2012-12-31	Demofirma DFØ-ID	9990	1	0	9990
2005-01-19	499	10001301	2005-01-01	2012-12-31	Demofirma DFØ-ID	9990	1	0	9990

Figur 2: Skjermbilde av organisasjonshiearki utvidet til daglig nivå, hvor DATO-feltet indikerer dagen.

Koble mot andre objekter

Organisasjonshierarkiet er et naturlig utgangspunkt for å bygge opp en analytisk modell og er derfor sentral i datamodellen. Den fungerer som en grunnstein man kan koble videre mot de andre objektene og slik bygge opp den ønskede analysemodellen. Datasettet kan med fordel utvides til daglig nivå på et tidlig tidspunkt i modelleringen, slik at man også får utvidet de resterende objektene i den videre sammenkoblingen.

Datasettet **stillingPlassering** kan koble feltet **ORGANISASJONSENHET** mot **ORGENHET_NR** i organisasjonshierarkiet for å lage lister over stillinger i de ulike organisasjonsenhetene per dag.

Tilsvarende kan datasettet **ansattPlassering** koble feltet **ORGANISASJONSENHET** mot **ORGENHET_NR** i organisasjonshiearkiet for å lage en liste over ansatte i organisasjonsenhetene per dag.

ansattPlassering har også feltet **STILLING_ID** som kan knyttes mot **stillingPlassering** for å populere stillingene med ansatte.

Deretter kan man fortsette å utvide datasettet med ulike dimensjoner innenfor de ulike områdene og gjøre tilpasninger basert på behov. For den resterende sammenkoblingen er det i all hovedsak feltet **ANSATTNUMMER** som benyttes og **STILLING_ID** for stillingsobjektene.