

# Udlæsningsmønstre til Fælleskommunalt Organisationssystem

## Dokumenthistorik

Dato	Ændring	Ændret af	Status	Beskrivelse
06.03.2019	Første udkast	Christian Buss Wennemoase		Første udkast  Dokumentet har været i review. Review kommentarer er ikke indarbejdet
11.05.2021	Opdateret med nye navn	Henrik Mynderup	Udgivet	Opdateret med nyt navn.

## 1 Anvendelse af Fælleskommunalt Organisationssystem

Nærværende dokument beskriver de mest almindelige udlæsningsmønstre for Fælleskommunalt Organisationssystem. Det antages at læseren er bekendt med registreringsmønstrene beskrevet i dokumentet "Anvisninger til anvendelse af Organisation"hg, som følges af kommunerne når de indlæser data i Fælleskommunalt Organisationssystem.

Dokumentet tager udgangspunkt i de behov en fagsystem-leverandør måtte have, for at tilgå kommunale organisationsdata, og beskriver en effektiv måde at udlæse disse data.

Bemærk at de fleste udlæsninger af data på nuværende tidspunkt kan være tidskrævende, og ikke bør udføres mens en slut-bruger sidder og venter i fagsystemet og venter på data, men i stedet bør udføres som batch-udlæsninger. Udlæsningsmønsteret for udlæsning af en enkelt bruger kan dog udføres på kort tid, og kan med fordel anvendes på login-tidspunkt, som en metode til at sikre friske data på den pågældende bruger.

## 2 Generelt om metoder på SOAP snitfladerne

Fælleskommunalt Organisationssystem udstiller SOAP snitflader til at udlæse organisationsdata. Disse snitflader består af et sæt af identiske operationer på hver objekttype (læs, list og søg), samt en sammenstykket operation ved navn "Frem søg Objekt Hierarki", der finder alle organisationsenheder der indgår i en kommunes organisationshierarki.

### 2.1 Objekttyper

De objekttyper der behandles i nærværende dokument er

- **Person.** Indeholder navn og eventuelt cpr på medarbejdere
- **Bruger.** Indeholder bruger-id på medarbejdere, og dette objekt er referencen til adgangsstyring for brugere
- **OrganisationEnhed.** Indeholder navne og relationer på organisationsenheder i organisationshierarkiet
- **Adresse.** Indeholder bl.a. telefonnumre, email adresse, web-adresser, postadresser, m.m.
- **OrganisationFunktion.** Indeholder rige relationer mellem objekter, fx engagementer mellem brugere og enheder

### 2.2 Operationer

De generelle operationer der er tilgængelige på hver objekttype er

- **Søg.** Kan anvendes til at foretage søgninger efter objekter der matcher specifikke søgekriterier. Søgeresultatet er referencer til objekter (UUID'er på objekter der matcher søgekriterierne)
- **Læs.** Kan anvendes til at udlæse et enkelt objekt. Input er et enkelt UUID.
- **List.** Kan anvendes til at udlæse mange objekter samtidig. Output er en liste af objekter med samme struktsur som Læs() operationen. Input er en liste af UUID'er.

Hvis man skal udlæse mere end ét objekt, er List operationen betydeligt hurtigere at anvende, end at kalde Læs operationen mange gange.

## 2.3 Sammenstilling af data

SOAP snitfladerne er indelt i forskellige fysiske endpoints, ét for hver objekttype, samt ét til tværgående organisationsdata. Det betyder at man skal lave et SOAP kald per objekttype, og sammenstille disse i fagsystemet, når data man har brug for går på tværs af objekttyper.

Dette medfører bl.a. at for at udlæse oplysninger om en medarbejder (fx navn, bruger-id og email adresse), skal man lave flere SOAP kald (til hhv Person, Bruger og Adresse), og sammenstille dem i fagsystemet for at danne sig et samlet billede af medarbejderen.

Scenarierne i næste afsnit beskriver hvordan man gør dette effektivt, uden at skulle foretage alt for mange SOAP kald.

## 3 Scenarier

### 3.1 Udlæs organisationshierarki

En gennemsnitskommune har ca 400-600 organisatoriske enheder, hvor nogle større kommuner har betydeligt flere. Det er muligt at anvende en kombination af Søg og List operationer til at udlæse alle enheder, men en mere effektiv metode findes til formålet – operationen Frem søg Objekt Hierarki, der findes på SOAP snitfladen OrganisationSystem.

Denne metode finder alle de enheder der indgår i organisationshierarkiet, og returnerer dem som fulde objekter, og ikke referencer. Ved at anvende metoden sikres både at man kun får de enheder ud som indgår i enhedshierarkiet (så man ikke unødigt udlæser enheder der af den ene eller anden grund ikke indgår i kommunens enhedshierarki), og man behøves ikke kombinere to SOAP kald (søg+læst). Frem søg Objekt Hierarki er samtidig også lidt hurtigere end de to andre operationer.

Man skal dog bemærke at Frem søg Objekt Hierarki ikke filtrerer enheder bort der har tilstanden Inaktiv, så disse skal man se bort fra i udlæsningen af data. Den manglende filtrering på inaktive objekter er en kendt fejl, som forventes løst primo Q1, 2019.

Frem søg Objekt Hierarki (på lige fod med søg operationen) udlæser 500 enheder ad gangen, så man skal paginere hen over resultat, så typisk ender man med at skulle foretage 1-2 SOAP kald for at udlæse alle relevante enheder i en given kommune.

Eksempel input til Frem søg Objekt Hierarki, med paginering angivet, kan se ud som vist nedenfor

```
<Body>
  <FremsoegobjekthierarkiRequest>
    <FremsoegObjekthierarkiInput>
      <FoersteResultatReference>0</FoersteResultatReference>
      <MaksimalAntalKvantitet>500</MaksimalAntalKvantitet>
    </FremsoegObjekthierarkiInput>
  </FremsoegobjekthierarkiRequest>
</Body>
```

I kaldet hentes 500 OrganisationEnheder (MaksimalAntalKvantitet), startende fra OrganisationEnhed nummer 0 (FoersteResultatReference). I det efterfølgende kald vil man ændre FoersteResultatReference til 500, for at udlæse fra OrganisationEnhed nummer 500. Bemærk at forsøg på at udlæse mere end 500 OrganisationEnheder ad gangen vil man modtage en fejlmeddelelse.

Da organisationshierarkiet for en kommune ikke ændrer sig hele tiden, kan man med fordel implementere denne udlæsning som et batch-job, der afvikles et par gange om dagen, for at sikre at man altid har et opdateret organisationhierarki tilgængelig.

### 3.2 Udlæs enkelt medarbejder

Udlæsning af en enkelt medarbejder kan udføres med en række SOAP kald, der kan gennemføres hurtigt nok til at det kan gøres i forbindelse med at en bruger logger på fagsystemet. På den måde kan man hente relevante supplerende oplysninger om brugeren (navn, cpr, bruger-id, telefonnummer, email, organisatorisk indplacering, osv).

Når der er tale om udlæsning af én medarbejder, er det begrænset hvor mange optimeringer der kan udføres – man kan dog altid udelade kald som vedrører data ens fagsystem ikke har brug for. Nedenstående kald-sekvens viser hvordan man udlæser samtlige organisatoriske data på en medarbejder på den mest effektive måde.

Når en bruger logger på ens fagsystem, er udgangspunktet at man har UUID'et på medarbejderens Bruger objekt i Organisation (fremgår af Serial feltet i SAML tokenet fra støttesystemet Adgangsstyring).

1. Der laves et kald til Læs operationen på Bruger snitfladen. Dette giver os adgang til følgende oplysninger
  - Medarbejderens bruger-id
  - En liste af reference til Adresse objekter (UUID'er på disse objekter)
  - En reference til et Person objekt (UUID'et på et Person objekt)
2. Der laves et kald til Læs operationen på Person snitfladen. Dette giver os adgang til følgende oplysninger
  - Medarbejderens navn
  - Medarbejderens CPR nummer (hvis fagsystemet har adgang til dette felt, og kommunen har indlæst det på medarbejderen – ellers er feltet tomt)
3. Der laves et kald til List operationen på Adresse snitfladen. Input hertil er de UUID'er som Bruger objektet har referencer til (man kan evt udvælge de UUID'er som man ønsker at udlæse – man kan se typen af adresse på referencen). Dette giver os adgang til følgende oplysninger
  - Den faktiske adresse-værdi (fx telefonnummeret, emailadressen, osv)
4. Der laves et kald til Søg operationen på OrganisationFunktion snitfladen. Input til denne søgning er en reference til Bruger objektet, samt en angivelse af type, hvor man beder om at få udlæst OrganisationFunktioner af typen 'Enhedsrolle'. Output fra dette kald er en liste af UUID'er der peger på OrganisationFunktion objekter, der repræsenterer engagementer mellem medarbejderen og OrganisationEnhed objekter (dvs medarbejderens organisatoriske indplaceringer).
5. Der laves et kald til List operationen på OrganisationFunktion snitfladen. Input til denne operation er de UUID'er som blev returneret fra Søg. Output er en liste af OrganisationFunktion objekter, hvor hvert objekt indeholder oplysninger om
  - Navnet på den organisatoriske indplacering (dette kunne være medarbejderens stillingsbetegnelse, men da der er tale om et fritext-felt, kan der stå hvad som helst)
  - En reference til den OrganisationEnhed som medarbejderen er indplaceret i
6. Endeligt laves et kald til List operationen på OrganisationEnhed snitfladen. Input er de UUID'er som fremkom fra udlæsningen af OrganisationFunktion objekter, og output indeholder oplysninger om den organisatoriske enhed, herunder
  - Enhedens navn

- En reference til den enhed der ligger over denne enhed i enhedshierarkiet (disse referencer kan man bruge til at bygge en egentlig træ-struktur over enhederne efterfølgende).

Hvis man har cachet de organisatoriske enheder, er det ikke nødvendigt at udføre det sidste kald, så kan man blot slå op i de cachede data for at finde oplysninger om enheden. Worst-case, så skal der udføres 6 SOAP kald for at udlæse alle relevante oplysninger på den ene medarbejder.

Alt efter ens behov, kan man udelade nogle af disse kald i udlæsningen af organisationsdata på medarbejderen.

### 3.3 Udlæs alle medarbejdere fra én enhed

En mere effektiv måde at udlæse oplysninger om medarbejdere på, er ved at udføre en bulk-udlæsning af flere medarbejdere samtidig. Man kunne fx tage udgangspunkt i alle medarbejdere i en given enhed på en gang. Denne bulk-udlæsning kunne man passende udføre samtidig med at man udlæser og vedligeholder ens cache af organisationshierarkiet.

I dette udlæsningsmønster gøres brug af fordele ved List operationen, der udføres **betydeligt** hurtigere end mange på hinanden følgende kald til Læs operationen. Det betyder dog at man i sin udlæsning skal udlæse objekttyper samlet (fx udlæse alle adresseobjekter samlet, og så flette adresseobjekter med bruger objekter efterfølgende). Det stiller lidt større krav til ens udlæsningslogik, da den skal holde styr på hvilke objekter har referencer til hvilke andre objekter, da selve sammenstillingen af objekter først sker efter alle data er udlæst.

Følgende kaldsekvens kan anvendes til at udlæse alle medarbejdere fra én enhed på en effektiv måde. Udgangspunktet er at man kender UUID'et på den enhed hvor man ønsker at udlæse medarbejdere for.

1. Der udføres et kald til Søg operationen på OrganisationFunktion snitfladen. Input til Søg operationen er UUID'et på OrganisationEnhed, samt en angivelse af at man kun ønsker OrganisationFunktioner af typen 'enhedsrolle'. Dette returner en liste af UUID'er på OrganisationFunktion objekter der repræsenterer engagementer mellem medarbejdere og den udvalgte enhed
2. Der udføres et kald til List operationen på OrganisationFunktion snitfladen. Input er de UUID'er der fremkom af den tidligere søgning. Output er OrganisationFunktion objekter, der indeholder følgende oplysninger
  - Navnet på den organisatoriske indplacering (dette kunne være medarbejderens stillingsbetegnelse, men da der er tale om et fritekst-felt, kan der stå hvad som helst)
  - En reference til det Bruger objekt som er relateret til enheden
3. UUID'erne på Bruger referencen i alle OrganisationFunktioner samles, og anvendes i et kald til List operationen på Bruger snitfladen. Output herfra er en liste af Bruger objekter, der indeholder
  - Medarbejderens bruger-id
  - En liste af reference til Adresse objekter (UUID'er på disse objekter)
  - En reference til et Person objekt (UUID'et på et Person objekt)
4. UUID'erne på Person referencen i alle Bruger objekterne samles, og anvendes i et kald til List operationen på Person snitfladen. Output herfra er en liste af Person objekter, der indeholder
  - a. Medarbejderens navn
  - b. Medarbejderens CPR nummer (hvis fagsystemet har adgang til dette felt, og kommunen har indlæst det på medarbejderen – ellers er feltet tomt)

5. Der laves et kald til List operationen på Adresse snitfladen. Input hertil er de UUID'er som alle Bruger objektet har referencer til (man kan evt udvælge de UUID'er som man ønsker at udlæse – man kan se typen af adresse på referencen). Dette giver os adgang til følgende oplysninger
  - a. Den faktisk adresse-værdi (fx telefonnummeret, emailadressen, osv)

Endeligt sammenstilles alle de udlæste data, hvilket kræver at man undervejs har holdt styr på hvilke objekter har referencer til hvilke andre objekter (UUID referencerne).

### 3.4 Fuld synkronisering af organisationsdata

En fuld synkronisering af organisationsdata kan udføres ved at kombinere udlæsningen af organisationshierarkiet, med en udlæsning af medarbejdere per enhed. Man kan optimere videre endnu, ved at først udlæse alle enheder, og dernæst påbegynde udlæsningen af alle medarbejdere i én stor bulk-operation. Her skal man dog være opmærksom på at Søg operationerne kun returnerer 500 UUID'er, og det er nødvendigt at paginere i søge-resultatet hvis man anvender denne strategi.

Man kan også, når man udfører bulk-operationer på rigtigt mange objekter, opleve lange svartider, som kan resultere i timeouts hvis man ikke øger den maksimale netværkstimeout i ens egen kode.

En udlæsning af medarbejdere per enhed, er formodentligt et fornuftigt trade-off mellem hurtig og effektiv udlæsning af data, og netværks-stabilitet under udlæsningen.

## 4 Optimeringstricks

De to mest effektive optimeringer i udlæsningen af data, er at anvende List operation på mange objekter samtidig, frem for at anvende Læs operationen, og så kun at udlæse data man har brug for.

Mht det sidste, så kan man anvende yderligere filtre i ens søge-operationer, til at sikre at man kun modtager referencer til objekter som er relevante for ens fagsystem.

### 4.1 Filtrering på it-system anvendelse

En af registreringsmønstrene for indlæsning af data, er opmærkning af organisatoriske enheder med it-system anvendelse. Man bør starte med at sikre at ens kommunale kunder anvender dette registreringsmønster inden man gør brug af det til filtrering. Under antagelse af at kommunen har registreret hvilke enheder der gør brug af ens fagsystem, så man kan udføre følgende filtrering af data inden man begynder udlæsningen af medarbejdere og deres organisatoriske data

For at kunne udføre denne filtrering skal man kende UUID'et på ens eget fagsystem.

1. Udfør et kald til Søg operationen på OrganisationFunktion snitfladen. Som input angives at man søger efter OrganisationFunktioner af typen 'systemrolle', hvor der er en relation til ItSystemet med et UUID der matcher ens eget fagsystem. Output herfra er en liste af UUID'er på OrganisationFunktioner der matcher søgekriterierne
2. Udfør et kald til List operationen på OrganisationFunktion snitfladen. Som input angives de UUID'er som blev returneret af ovenstående søgning. Output er en liste af OrganisationFunktion objekter, der indeholder
  - Referencer til OrganisationEnhed objekter der anvender fagsystemet

Hvis man tidligere har udlæst det fulde enhedshierarki, kan man nu opmærke ens lokale cache med hvilke enheder der gør brug af ens fagsystem, og kan så nøjes med at udlæse medarbejdere fra disse OrganisationEnheder.

Hvis man endnu ikk har udlæst det fulde enhedshierarki, kan man også vælge at bruge denne liste af UUID'er som input til List operationen på OrganisationEnhed snitfladen, for at udlæse oplysninger om enhederne.