

Flow for data fra register til anvender

Oprindelig kilde	Datafordeleren
Forfatter	Datafordeleren
Oprettet	Jan 08, 2021
Version	1.1
Ændret	Oct 30, 2022
Sidehistorik	• Se sidehistorik

Introduktion

Her gives en kort beskrivelse af dataflow mellem register og Datafordeleren samt de tekniske valideringer der gennemføres som en del af flowet. Efterfølgende beskrives flow fra Datafordeler til anvender.

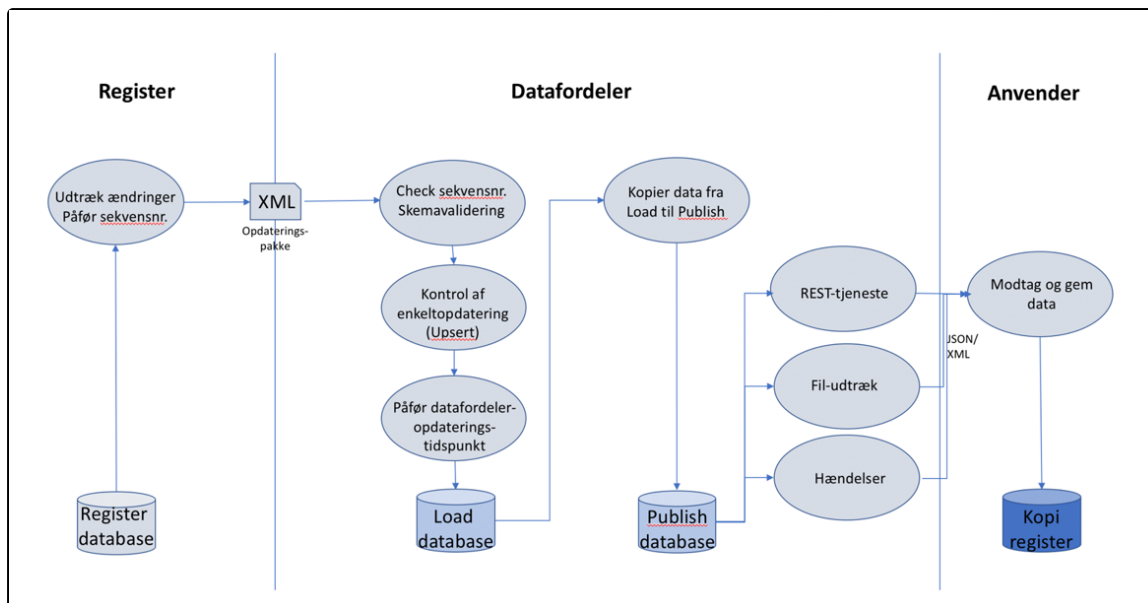
Datafordeleren gennemfører ikke nogen forretningsmæssig validering af data. Dette omfatter også at der ikke er nogen validering af bitemporale attributter, ud over at de indholdsmæssigt skal overholde det aftalte date-time format.

- [Dataflow fra register til anvender](#)
- [Register](#)
- [Indlæsningsflow for Datafordeleren](#)
- [Valideringer ved udstilling](#)

Dataflow fra register til anvender

[Til toppen](#)

Det samlede flow er illustreret i nedenstående figur



Beskrivelse af flow for data fra register datafordeler anvender, herunder implementerede kontrol/validerings-mekanismer mellem register og Datafordeler

Register

[Til toppen](#)

Register udtrækker ændringer fra registres database og skriver disse til en opdateringspakke der sendes til Datafordeleren.

Hver ændring der sendes til Datafordeleren markeres som en af følgende:

- Update: Angiver at det er en eksisterende instans på Datafordeleren der skal opdateres. Vil typisk være lig med en række i Datafordelerens database. Det bemærkes her, at ved opdatering af bitemporale objekter sker der både en "Update" og en "Insert". Dvs "Update" af eksisterende række hvor registrering til sættes og "Insert" af en ny række med de nye dataværdier.
- Insert: Angiver at der skal indsættes en ny instans i Datafordelerens database. Vil typisk være lig med en ny række.
- Delete: Angiver at en eksisterende instans på Datafordeleren skal slettes. Dette gøres som en softdelete, som er en Datafordelerfunktion, hvor en række markeres som slettet og ikke længere kan ses eller tilgås af anvendere i tjenester. Hverken REST eller filudtræk.

Når alle ændringer der skal sendes til Datafordeleren er udtrukket, samles disse i en opdateringspakke. Hver opdateringspakke tildeles et sekvensnummer, dvs. et nummer fra en fortløbende nummerserie der er aftalt mellem Register og Datafordeleren. Det er gennem sekvensnummer at Register angiver hvilken rækkefølge opdateringspakker indlæses på Datafordeleren.

Afslutningsvis sendes opdateringspakke til Datafordeleren. Teknisk sker dette gennem en SOAP eller SFTP-snitflade udstillet af Datafordeleren. På snitfladen anvendes altid XML.

Kort opsummeret sker følgende, jf. ovenstående:

1. Register udtrækker ændringer og angiver type på dataændring
2. Register genererer opdateringspakke med næste sekvensnummer
3. Register sender opdateringspakke til datafordeler

Når Datafordeler har modtaget en opdateringspakke indlæses data først i Master Load databasen og ved succesfuld indlæsning, kopieres data til Publish-databaser. Det er fra Publish-databaser at tjenester udstillet på Datafordeleren henter data.

Totalindlæsninger håndteres principielt på samme måde, dog med den forskel at alle data slettes fysisk i tabellerne (både i Load og Publish databaserne), hvorved alle data håndteres som Insert.

Indlæsningsflow for Datafordeleren

[Til toppen](#)

Når Datafordeleren har modtaget en opdateringspakke, gennemføres følgende før og under indlæsning af data i Master Load.

1. Datafordeleren registrerer opdateringspakken som klar til indlæsning og flytter til arbejdsfolder
2. Opdateringspakkens sekvensnummer valideres (er det næste pakke der skal indlæses)
3. Der gennemføres skemavalidering af hele opdateringspakken
4. Datafordeleren starter en transaktion, der favner følgende
 - a. Load-database opdateres med data fra opdateringspakke. Hver række påføres et Datafordeler opdateringstidspunkt
 - b. Kontrol af enkeltopdateringer (~~gælder ikke alle registre~~)
 - i. Ændringstype "Insert": Hvis der allerede findes en række, fejler indlæsning
 - ii. Ændringstype "Update": Hvis række ikke findes, dvs. der ikke kan gennemføres en opdatering fejler indlæsning
 - iii. Ændringstype "Delete": Hvis række ikke findes fejler indlæsning (soft-delete)
 - c. Korrektion af fejl
 - i. Fejl løses typisk ved at opdateringspakken tilrettes af register og fremsendes på ny med samme sekvensnummer (bemærk af andre opdateringer med nyere sekvensnummer afventer genfremsendelse af opdateringspakke)
 - ii. Registre kan også vælge at anvende en funktionalitet i Datafordeleren, der kaldes Upserts, hvorved af en række fejlscenarier håndteres af Datafordeleren under indlæsning. Upsert har følgende funktionalitet:
 1. Insert-fejl: Situationer hvor række findes i forvejen og insert derfor fejler, håndteres ved at Insert konverteres til en opdatering (Update)
 2. Updatefejl: Situationer hvor der forsøges opdateret en række der ikke findes, håndteres ved at opdateringen konverteres til en oprettelse (Insert)
 - d. Når alle data er indlæst i Load-databasen, kopierer data til Publish-databasen
 - e. Indlæsningsjob markeres færdig
 - f. Sekvensnummer inkrementeres så næste opdateringspakke kan indlæses
5. Transaktionen slutes og alle ændringer commites i databaserne

Valideringer ved udstilling

[Til toppen](#)

Der gennemføres kun ganske få validering for REST-tjenester. Det sikres at data opfylder fremsøgningslogik.

Ved hændelser gennemføres der ikke nogen valideringer af data.

Ved filudtræk udføres der enkelte valideringer, afhængig af definition af filudtræk:

- Ved totaludtræk gennemføres ingen valideringer
- Ved delta-udtræk hvor der er angivet Since Previous, valideres det at udtræk som udgangspunkt indeholder alle rækker, hvor datafordeler-opdateringstidspunkt er nyere end sidste gang delta-udtrækket blev kørt. [Læs mere om Deltaudtræk og dataintegritet](#)
- Ved brugerdefinerede udtræk sikres, at data opfylder de opsatte kriterier. Validering sker på objektniveau, men ikke på tværs af objekter.

Der er ingen logik på Datafordeleren, der holder styr på hvilke filer en anvender har modtaget.

Datafordeleren håndterer også oprydning i gamle udtræksfiler, da disse slettes i forhold til det interval registret har defineret for det enkelte filudtræk. Dette er typisk 30 dage. Det er ikke muligt at genkøre et deltaudtræk, hvis man som anvender ikke har fået hentet filen og den er blevet slettet af Datafordelerens oprydning.
