

# Hændelser på Datafordeleren

|                         |                                                                                   |
|-------------------------|-----------------------------------------------------------------------------------|
| <b>Oprindelig kilde</b> | Datafordeleren                                                                    |
| <b>Forfatter</b>        | Datafordeleren                                                                    |
| <b>Oprettet</b>         | Jun 11, 2019                                                                      |
| <b>Version</b>          | 1.0                                                                               |
| <b>Ændret</b>           | Oct 30, 2023                                                                      |
| <b>Sidehistorik</b>     | <ul style="list-style-type: none"><li>• <a href="#">Se sidehistorik</a></li></ul> |

---

Denne side har til hensigt at give en introduktion til begrebet hændelser, som det er implementeret på Datafordeleren.

I de følgende afsnit introduceres først den overordnede arkitektur, hvorefter hændelsesbegrebet og hændelsesbeskeder beskrives. Herefter beskrives objekter og deres repræsentation i databaser med dobbelthistorik, dataopdateringer samt abonnementer. Et gennemgående eksempel fremstiller det samlede flow fra register til anvender, og endelig kommer vi ind på mulige anvendelser af hændelser.

- For oplysninger om hvilke hændelser og data, der er tilgængelige på Datafordeleren, brug [Dataoversigten](#).
- Se [Vilkår for anvendelse](#) for de enkelte registre på Datafordeler.dk.
- Læs mere generel information om hændelsesbaseret eller beskedbaseret arkitektur (Event-driven architecture, EDA) på [Arkitekturguiden - Hvad er EDA?](#)
- Få en guide til opsætning af abonnementer på [Guide til hændelser på Datafordeleren](#), eller [Guide til hændelser på Selvbetjeningen](#).

- 
- Overordnet arkitektur
  - Hændelsesbegrebet
    - Forretningsmæssige og datanære hændelser
    - Hændelsesbeskeder
  - Objektbegrebet
    - Databaserepræsentation
  - Dataopdateringer
  - Abonnementer
    - PUSH af hændelser
      - Kodeeksempel til modtagelse af OData PUSH hændelser
    - PULL af hændelser
      - Paging
      - Kodeeksempel udstilling af hændelser på kort
    - Kvittering
  - Eksempel på hændelse fra opdatering til udsendelse
  - Anvendelse af hændelser
    - Brugsscenarier
    - Hændelser versus deltafiler
    - Tidsdimensioner
  - Andre abonnementer
    - Abonnementer på filudtræk
    - Abonnement på Atom feeds
-

# Overordnet arkitektur

[Til toppen](#)

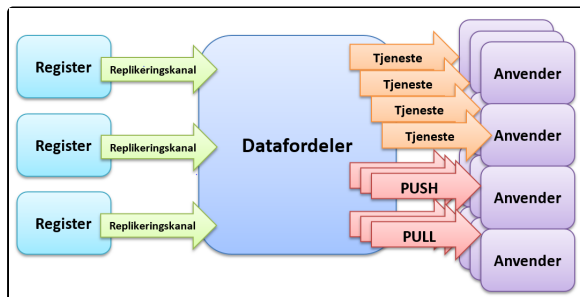
Nedenstående figur viser den overordnede arkitektur for distribution af data via Datafordeleren.

Data bliver oprettet og vedligeholdt i registre og løbende kopieret til Datafordeleren via replikeringskanaler.

Anvendere kan hente data fra Datafordeleren gennem forskellige former for tjenester, herunder filudtræk.

Udover selve data, distribuerer Datafordeleren også hændelsesbeskeder. For at modtage hændelsesbeskeder skal en anvender oprette et abonnement på den type af hændelsesbeskeder, der ønskes.

Ved opsætning af abonnementer er der mulighed for at filtrere yderligere i hændelsesbeskederne på baggrund af deres indhold, og der vælges mellem at få tilsendt hændelsesbeskeder (PUSH scenarie) eller selv hente hændelsesbeskeder på Datafordeleren (PULL scenarie).



---

## Hændelsesbegrebet

[Til toppen](#)

En hændelse er, når der sker noget. I et register betyder det, at der sker en ændring af data. En hændelse kan således være resultatet af et trin i en forvaltningsproces (også omtalt som en forretnings-proces), som opdaterer data i et register.

I daglig tale skelner man ofte ikke skarpt mellem hændelser og hændelsesbeskeder, men ofte benyttes termene synonymt. Denne side tilstræber at afspejle den typiske sprogbrug omkring hændelser.

---

## Forretningsmæssige og datanære hændelser

Datafordeleren giver registermyndigheder mulighed for at udsende to former for hændelser – forretningsmæssige og datanære.

Et register kan danne hændelsesbeskeder på baggrund af forretningsmæssige hændelser fra forvaltningsprocesser i registeret og sende dem til Datafordeleren via en særskilt replikeringskanal til videre distribution til abonnenter. Dette er forretningsmæssige hændelser.

Datanære hændelser dannes af Datafordeleren på baggrund af dataopdateringer modtaget fra et register på replikerings-kanaler. Dannelse og distribution af en hændelsesbesked er derfor bundet til en opdatering af data. Herved undgår man situationer, hvor man fx udsender (forretningsmæssige) hændelsesbeskeder om ændringer i data, men alligevel ikke får gennemført en opdatering af data, som er tilgængelige på Datafordeleren.

For en anvender er der i princippet ikke forskel på disse former for hændelser. De har samme format, som beskrives i et senere afsnit, mens indholdet af hændelsesbeskederne selvsagt kan variere.

I det følgende omtaler vi datanære hændelser.

---

## Hændelsesbeskeder

En hændelsesbesked har en beskedtype. For datanære hændelser er beskedtyper dannet ved at sammen-sætte objekttype og operation på objektet. Operationen på objektet kan være Create, Update eller Delete. Et eksempel på en beskedtype er KommuneinddelingUpdate, der benyttes til Update-hændelser for objekttypen Kommuneinddeling.

Hændelsesbeskeder har et fast format med hensyn til struktur, men vil være tilgængelige i XML og JSON.

Nedenstående figur viser den overordnede struktur i hændelsesbeskeder.



En hændelsesbesked indeholder en beskedkuvert og beskeddata. Beskedkuverten indeholder blandt andet filtreringsdata, der kan benyttes til filtrering af hændelsesbeskeder i forbindelse med abonnementer. Filtreringsdata indeholder en eller flere objektregistreringer, der indeholder feltet Objekthandling. Praksis omkring brug af dette felt er, at registrene kan benytte det til at overlevere ekstra information omkring hændelsen, som ikke er direkte afledt af selve data. Det kan fx være en label for den forretningsmæssige årsag til hændelsen eller en liste med navne på de felter, der er ændret ved dataopdateringen. Ved opsætning af abonnementer kan disse ekstra informationer i feltet Objekthandling udnyttes til filtrering.

---

## Objektbegrebet

[Til toppen](#)

For at forstå dannelsen af datanære hændelser på Datafordeleren er det nyttigt at forstå, hvordan objekter repræsenteres i databaser og kopieres fra registre til Datafordeleren via replikeringskanaler.

I datamodellering skelner man ofte mellem konkrete (fysiske eller konceptuelle) objekter /entiteter og deres repræsentation i form af forvaltnings- eller forretningsobjekter. På denne side omtaler vi for nemheds skyld blot forvaltnings- eller forretningsobjekter som objekter, idet fokus ikke er på modellering og samspil med virkeligheden, men udelukkende på repræsentationen og operationer i systemerne.

---

## Databaserepræsentation

Objekter repræsenteres i databasen med rækker. Det betyder, at ét objekt og dets historik er repræs-en-teret i databasen med en eller flere rækker. I Grunddataprogrammet indeholder data bitemporale egenskaber (dobbelthistorik). Det vil sige at en række indeholder information om registreringstid og virkningstid i form af tidsintervaller. Registreringstid angiver, hvornår informationerne om objektet var kendt. Virkningstid angiver, hvornår informationerne er gyldige i forhold til virkeligheden. Registrerings- og virkningstid er i databasen angivet som tidsintervaller i hver række med felterne registreringFra, registreringTil, virkningFra og virkningTil. Tidsintervallerne kan være åbne, da man ikke nødvendigvis har en slutdato.

En ændring i et objekt kan betyde ændringer og oprettelse af flere rækker i databasen. Oprettelse af et objekt bevirker oprettelse af en ny databaserække. Ændring af et objekt med bitemporale egenskaber foretaget til tiden t kan f.eks. bevirke, at der oprettes en ny databaserække med registreringFra = t og registreringTil = null, mens databaserækken for seneste forekomst af objektet får sat registreringTil = t.

Ved ændringer i objekter oprettes normalt flere databaserækker, der afspejler objektet til forskellige tider, det vil sige forskellige intervaller i virkningstiden. Når man vedligeholder bitemporale egenskaber, vil man normalt slette objekter ved at opdatere dem, således at virkningstiden indikerer, at det ikke længere eksisterer. Sletning af objekter oprettet ved en fejl, kan dog udføres ved at sætte registreringTil = t, dvs. man udfører en afregistrering (eller logisk sletning) af den aktuelle databaserække.

Et eksempel på oprettelse og efterfølgende fejlrettelse i et objekt er illustreret i nedenstående tabeller. Her oprettes et objekt først ved at oprette en ny række, hvorefter objektet opdateres ved at ændre registreringstid i rækken (registreringTil udfyldes) og oprette en ny række med det rettede data. I eksemplet er der blot tale om en fejlrettelse, dvs. der rettes blot en stavfejl og der ændres ikke i virkningstid.

#### Oprettelse af objekt – ny række i databasen:

| ID | UUID | Data     | Reg. Fra   | Reg. Til | Virk. Fra  | Virk. Til |
|----|------|----------|------------|----------|------------|-----------|
| 1  | xxxx | Grøndahl | 01-09-2016 | -        | 01-10-2016 | -         |

#### Fejlrettelse i objekt – ændring i eksisterende række samt ny række i databasen:

| ID | UUID | Data     | Reg. Fra   | Reg. Til   | Virk. Fra  | Virk. Til |
|----|------|----------|------------|------------|------------|-----------|
| 1  | xxxx | Grøndahl | 01-09-2016 | 07-09-2016 | 01-10-2016 | -         |
| 2  | xxxx | Grøndal  | 07-09-2016 | -          | 01-10-2016 | -         |

Hvis der er tale om en decideret ændring af stavemåde, vil det give anledning til to nye rækker for at holde styr på virkningstiden for den gamle og den nye stavemåde. Dette er den almindelige fremgangsmåde for ændringer, og et eksempel er illustreret i nedenstående tabeller. Her oprettes et objekt først ved at oprette en ny række. En opdatering af objektet, som er en ændring af stavemåde, resulterer i, at den hidtidige registrering (række) afregistreres ved at udfylde registreringTil, og at der oprettes to nye rækker for at holde styr på virkningstiden for den gamle og den nye stavemåde.

#### Oprettelse af objekt – ny række i databasen:

| ID | UUID | Data    | Reg. Fra   | Reg. Til | Virk. Fra  | Virk. Til |
|----|------|---------|------------|----------|------------|-----------|
| 3  | yyyy | Aabakke | 01-11-2016 | -        | 01-12-2016 | -         |

#### Opdatering af objekt – ændring i eksisterende række samt to nye rækker i databasen:

| ID | UUID | Data    | Reg. Fra   | Reg. Til   | Virk. Fra  | Virk. Til  |
|----|------|---------|------------|------------|------------|------------|
| 3  | yyyy | Aabakke | 01-11-2016 | 04-01-2017 | 01-12-2016 | -          |
| 4  | yyyy | Aabakke | 04-01-2017 | -          | 01-12-2016 | 01-02-2017 |
| 5  | yyyy | Åbakke  | 04-01-2017 | -          | 01-02-2017 | -          |

Attributter med status, der angiver, hvor objektet er i sin livscyklus, samt registrerings- og virkningsaktører er ikke taget med i eksemplerne. Ligeledes er der heller ikke vist eksempler på ændringer, der omfatter flere objekter eller flere virkningsperioder.

For en nærmere gennemgang af bitemporalitet, som det benyttes i Grunddataprogrammet, henvises til [Datafordeleren - introduktion til bitemporalitet](#).

Krav til bitemporalitet i datamodellering beskrevet i [Modelreglerne for Grunddata](#).

---

[Til toppen](#)

## Dataopdateringer

Registrene overfører data via replikeringskanaler til Datafordeleren i XML. Der findes også replikeringskanaler til overførsel af rasterdata, fx skærmkort i billedfiler, men her betragter vi kun registerdata i relationelle databaser. Når et register sender dataopdateringer til Datafordeleren, svarer de til opdateringer af rækker i databasen. En databaserække kan oprettes, opdateres og slettes. Når man vedligeholder bitemporale egenskaber, vil man ved opdateringer kun opdatere feltet registrering. Til, det vil sige foretage en afregistrering af databaserækken. Derudover vil man ikke slette databaserækker.

Registeret har mulighed for at lade en dataopdatering (operation på en databaserække) danne en datanær hændelse. Registeret kan trigge en hændelse med en type svarende til operationen på objektet set fra et forvaltningsmæssigt synspunkt, og beskedtypen følger således ikke nødvendigvis dataopdateringens type. En oprettelse af en databaserække kan trigge en Update-hændelse. En opdatering af en databaserække kan trigge en Delete-hændelse. En dataopdatering behøver ikke nødvendigvis danne en datanær hændelse. Ved en opdatering af et objekt, som omfatter oprettelse og opdatering af flere rækker i databasen, kan registeret således lade en enkelt af dataopdateringerne trigge dannelsen af en datanær hændelse svarende til den forretningsmæssige ændring af objektet, mens de resterende dataopdateringer ikke giver anledning til hændelser.

Anvendere vil kun se de hændelsesbeskeder, der bliver dannet til abonnementerne, og behøver ikke forholde sig til de ændringer og oprettelser af rækker i databasen, der ligger til grund. Registeret har taget stilling til dannelsen af de datanære hændelser, og anvenderne får oplysninger gennem hændelsesbeskeder og selve data udstillet i tjenester, herunder filudtræk.

Se mere om hvor ofte registrene sender dataopdateringer til Datafordeleren på [Dataopdateringer](#).

---

[Til toppen](#)

## Abonnementer

En anvender skal have oprettet et abonnement, før det er muligt at få tilsendt eller hente hændelsesbeskeder. Et abonnement er tilknyttet en adgang til data (gennem en tjenestebruger), og begge dele oprettes i Selvbetjeningen.

Et abonnement kan sættes op til, at Datafordeleren enten kan sende hændelsesbeskeder (PUSH) til en anvender eller anvenderen selv kan hente hændelsesbeskeder (PULL) ved at kalde en tjeneste på Datafordeleren svarende til det opsatte abonnement.

Det er muligt at opsætte et filter til et abonnement, således at abonnementet kun får hændelsesbeskeder, der opfylder betingelserne i filteret.

Læs mere om oprettelsen af abonnementet på [Guide til hændelser på Datafordeleren](#), eller [Guide til hændelser på Selvbetjeningen](#).

---

[Til toppen](#)

## PUSH af hændelser

For at Datafordeleren kan sende hændelsesbeskeder til en anvender, skal anvenderen implementere en tjeneste, der kan modtage hændelsesbeskeder. Tjenesten skal følge OData v4.0 protokollen og have et HTTPS-end-point, som Datafordeleren vil tilgå med et FOCES certifikat. Endpoint for tjenesten skal angives ved oprettelse af abonnement, når man vælger levering ved PUSH.

Ved PUSH bliver hændelsesbeskederne sendt til anvenderen umiddelbart efter, de er dannet. Hvis det ikke lykkes at aflevere hændelsesbeskederne, vil Datafordeleren forsøge igen én gang i timen, indtil hændelsesbeskederne er leveret, eller der er gået 2 måneder, hvorefter abonnementet vil blive deaktiveret og hændelsesbeskederne slettet.

Tjenesten, der skal udstilles af anvender, skal understøtte POST og batch request.

Du skal benytte et certifikat for at modtage hændelser via PUSH fra Datafordeleren - [hent den offentlige nøgle til det certifikat](#).

Content-Type for request (afsendt af Datafordeler) vil være application/json, hvor strukturen er:

```
DataDistributor.Event
{
  Id (integer, optional): Id,
  Format (string, optional): Format, // JSON or XML
  Body (string, optional): Body // encoded XML or JSON string
}
```

**Eksempel på struktur af REQUEST i JSON format:**

```
{
  "@odata.type": "#DataDistributor.Event",
  "Body": "{ \"sampleJsonField\": \"sampleValue\" }",
  "Format": "JSON",
  "Id": 1
}
```

**Eksempel på struktur af REQUEST i XML format:**

```
{
  "@odata.type": "#DataDistributor.Event",
  "Body": "<?xml version='1.0'> <Event>
    <sampleXmlField>sampleValue</sampleXmlField> </Event>",
  "Format": "XML",
  "Id": 1
}
```

Følgende eksempler viser request, som Datafordeleren benytter til at pushe hændelser til anvenderens tjeneste med POST samt response, som tjenesten skal levere i henhold til OData-specifikationen.

### Eksempel på en enkelt hændelsesbesked - REQUEST

```
POST http://host:1234/odata/Events HTTP/1.1
OData-Version: 4.0
OData-MaxVersion: 4.0
Content-Type: application/json;odata.metadata=minimal
Accept: application/json;odata.metadata=minimal
Accept-Charset: UTF-8
User-Agent: Microsoft ADO.NET Data Services
Host: test:1234
Content-Length: 110
Expect: 100-continue
Connection: Keep-Alive

{
  "@odata.type": "#DataDistributor.Event", "Body":
    "{ \"Hændelsesbesked\": {
      \"beskedversion\": \"1.0\",
      \"beskedId\": null,
      \"Beskedkuvert\": {
        \"Filtreringsdata\": {
          \"beskedtype\": \"MatrikulaerSagCreate\",
          \"beskedansvarligAktør\": null,
          \"tilladtModtager\": null,
          \"RelateretObjekt\": null,
          \"Objektregistrering\": [
            { \"registreringsaktør\": \"Anders Andersen\",
              \"registreringstid\": null,
              \"status\": \"Afsluttet\",
              \"objektansvarligAktør\": \"Geodatastyrelsen\",
              \"objektID\": \"ID20165\",
              \"objekttype\": \"MatrikulaerSag\",
              \"objekthandling\": null,
              \"opgaveemne\": \"52.20.05\",
              \"registreringsID\": \"5,0001-01-01T00:00:00.0000000\",
              \"Stedbestemmelse\": null
            }
          ],
          \"tværgåendeProces\": \"Diverse\"
        },
        \"Leveranceinformation\": {
          \"dannelsesetidspunkt\": null,
          \"transaktionsID\": null,
          \"kildesystem\": null,
          \"kildesystemIPAdresse\": null,
          \"kildesystemAkkreditiver\": null,
          \"sikkerhedsklassificering\": \"Ikke fortrolige data\",
          \"Leverancerute\": null
        },
        \"Modtagerhandling\": null
      },
      \"Beskeddata\": [
        { \"Objektreference\": { \"objektreference\": \"ID20165\" }, \"
Objektdata\": null }
      ]
    } }\",
  \"Format\": \"JSON\", \"Id\": 1
}
```

#### Eksempel på en enkelt hændelsesbesked - RESPONSE

```
HTTP/1.1 201 Created
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; odata.metadata=minimal;
charset=utf-8
Expires: -1
Location: http://test:1234/odata/Events(1)
Server: Microsoft-IIS/8.0
OData-Version: 4.0
X-AspNet-Version: 4.0.30319
X-SourceFiles: =?UTF-8?B?
QzpcUHJvamVjdHNCRGF0YWZvcmlbGVyXE1haw5cU29sdXRpb250ZXdcRGF0YURpc3R
yaWJ1dG9yLkV2ZW50c1xEYXRhRGlzdHJpYnV0b3IuRXZlbnRzLkNsaWVudFB1c2hTZX
J2aWNLXG9kYXRhXEV2ZW50cw==?=
X-Powered-By: ASP.NET
Date: Wed, 09 Dec 2015 12:49:34 GMT
Content-Length: 150
```

```
{
  "@odata.context": "http://test:1234/odata/$metadata#Events
/$entity", "Id": 1, "Format": "JSON", "Body":
  { "Hændelsesbesked": {
    "beskedversion": "1.0",
    "beskedId": null,
    "Beskedkuvert": {
      "Filtreringsdata": {
        "beskedtype": "MatrikulaerSagCreate",
        "beskedansvarligAktør": null,
        "tilladtModtager": null,
        "RelateretObjekt": null,
        "Objektregistrering": [
          { "registreringsaktør": "Anders Andersen",
            "registreringstid": null,
            "status": "Afsluttet",
            "objektansvarligAktør": "Geodatastyrelsen",
            "objektID": "ID20165",
            "objekttype": "MatrikulaerSag",
            "objekthandling": null,
            "opgaveemne": "52.20.05",
            "registreringsID": "5,0001-01-01T00:00:00.0000000",
            "Stedbestemmelse": null
          }
        ],
        "tværgåendeProces": "Diverse"
      },
      "Leveranceinformation": {
        "dannelsesetidspunkt": null,
        "transaktionsID": null,
        "kildesystem": null,
        "kildesystemIPAdresse": null,
        "kildesystemAkkreditiver": null,
        "sikkerhedsklassificering": "Ikke fortrolige data",
        "Leverancerute": null
      },
      "Modtagerhandling": null
    },
    "Beskeddata": [
      { "Objektreference": { "objektreference": "ID20165"}, "
Objektdata": null }
    ]
  } }
}
```

#### Eksempel på batch af hændelsesbeskeder - REQUEST

```
POST http://test:1234/odata/$batch HTTP/1.1
OData-Version: 4.0
OData-MaxVersion: 4.0
Content-Type: multipart/mixed; boundary=batch_e5961e6a-c65a-40d0-
a3b7-dd9653368f8e
Accept: multipart/mixed
Accept-Charset: UTF-8
User-Agent: Microsoft ADO.NET Data Services
Host: test:1234
Content-Length: 1280
Expect: 100-continue
Connection: Keep-Alive
```

--batch\_e5961e6a-c65a-40d0-a3b7-dd9653368fbe  
Content-Type: multipart/mixed; boundary=changeset\_45479556-c451-40c0-9dd6-69694b75a6db

--changeset\_45479556-c451-40c0-9dd6-69694b75a6db  
Content-Type: application/http  
Content-Transfer-Encoding: binary  
Content-ID: 1

POST http:// test:1234/odata/Events HTTP/1.1  
OData-Version: 4.0  
OData-MaxVersion: 4.0  
Content-Type: application/json;odata.metadata=minimal  
Accept: application/json;odata.metadata=minimal  
Accept-Charset: UTF-8  
User-Agent: Microsoft ADO.NET Data Services

```
{"@odata.type": "#DataDistributor.Event", "Body":
  {"Hændelsesbesked": {
    "beskedversion": "1.0",
    "beskedId": null,
    "Beskedkuvert": {
      "Filtreringsdata": {
        "beskedtype": "MatrikulaerSagCreate",
        "beskedansvarligAktør": null,
        "tilladtModtager": null,
        "RelateretObjekt": null,
        "Objektregistrering": [
          { "registreringsaktør": "Anders Andersen",
            "registreringstid": null,
            "status": "Afsluttet",
            "objektansvarligAktør": "Geodatastyrelsen",
            "objektID": "ID20165",
            "objekttype": "MatrikulaerSag",
            "objekthandling": null,
            "opgaveemne": "52.20.05",
            "registreringsID": "5,0001-01-01T00:00:00.0000000",
            "Stedbestemmelse": null
          }
        ],
        "tværgåendeProces": "Diverse"
      },
      "Leveranceinformation": {
        "dannelsesetidspunkt": null,
        "transaktionsID": null,
        "kildesystem": null,
        "kildesystemIPAdresse": null,
        "kildesystemAkkreditiver": null,
        "sikkerhedsklassificering": "Ikke fortrolige data",
        "Leverancerute": null
      },
      "Modtagerhandling": null
    },
    "Beskeddata": [
      { "Objektreference": { "objektreference": "ID20165" }, "
Objektdata": null }
    ]
  }},
  "Format": "JSON", "Id": 1}
```

--changeset\_45479556-c451-40c0-9dd6-69694b75a6db  
Content-Type: application/http  
Content-Transfer-Encoding: binary  
Content-ID: 2

POST http:// test:1234/odata/Events HTTP/1.1  
OData-Version: 4.0  
OData-MaxVersion: 4.0  
Content-Type: application/json;odata.metadata=minimal  
Accept: application/json;odata.metadata=minimal  
Accept-Charset: UTF-8  
User-Agent: Microsoft ADO.NET Data Services

```
{"@odata.type": "#DataDistributor.Event", "Body":
  {"Hændelsesbesked": {
    "beskedversion": "1.0",
    "beskedId": null,
    "Beskedkuvert": {
      "Filtreringsdata": {
        "beskedtype": "MatrikulaerSagCreate",
```

```

        "beskedansvarligAktør":null,
        "tilladtModtager":null,
        "RelateretObjekt":null,
        "Objektregistrering":[
            {
                "registreringsaktør":"Anders Andersen",
                "registreringstid":null,
                "status":"Afsluttet",
                "objektansvarligAktør":"Geodatastyrelsen",
                "objektID":"ID20166",
                "objekttype":"MatrikulaerSag",
                "objekthandling":null,
                "opgaveemne":"52.20.05",
                "registreringsID":"5,0001-01-01T00:00:00.0000000",
                "Stedbestemmelse":null
            }
        ],
        "tværgåendeProces":"Diverse"
    },
    "Leveranceinformation":{
        "dannelsesetidspunkt":null,
        "transaktionsID":null,
        "kildesystem":null,
        "kildesystemIPAdresse":null,
        "kildesystemAkkreditiver":null,
        "sikkerhedsklassificering":"Ikke fortrolige data",
        "Leverancerute":null
    },
    "Modtagerhandling":null
},
"Beskeddata":[
    {"Objektreference":{"objektreference":"ID20166"},"
Objektdata":null}
]
}}",
"Format":"JSON", "Id":2}
--changeset_45479556-c451-40c0-9dd6-69694b75a6db--
--batch_e5961e6a-c65a-40d0-a3b7-dd9653368fbc--

```

#### Eksempel på batch af hændelsesbeskeder - RESPONSE

```

HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: multipart/mixed; boundary=batchresponse_f75bc63a-8b9e-4614-b081-c910c850d22c
Expires: -1
Server: Microsoft-IIS/8.0
OData-Version: 4.0
X-AspNet-Version: 4.0.30319
X-SourceFiles: =?UTF-8?B?
QzpcUHJvamVjdHNCRGF0YWZvcmlbGVyXE1haw5cU29sdXRpb250ZXdcRGF0YURpc3RyaWJ1dG9yLkV2ZW50c1xEYXRhRG1zdHJpYnV0b3IuRXZlbnRzLkNsaWVudFB1c2hTZXJ2aWNLXG9kYXRhXCRiYXRjaA==?=
X-Powered-By: ASP.NET
Date: Wed, 09 Dec 2015 12:53:34 GMT
Content-Length: 1218

--batchresponse_f75bc63a-8b9e-4614-b081-c910c850d22c
Content-Type: multipart/mixed; boundary=changesetresponse_f78a5a5e-53d4-42af-a334-ab7001cbe4fb

--changesetresponse_f78a5a5e-53d4-42af-a334-ab7001cbe4fb
Content-Type: application/http
Content-Transfer-Encoding: binary
Content-ID: 1

HTTP/1.1 201 Created
Location: http://host:1234/odata/Events(1)
Content-Type: application/json; odata.metadata=minimal; charset=utf-8
OData-Version: 4.0

{
  "@odata.context":"http://test:1234/odata/$metadata#Events/$entity",
  "Id":1,
  "Format":"JSON",
  "Body":{
    "Hændelsesbesked":{
      "beskedversion":"1.0",
      "beskedId":null,

```

```

"Beskedkuvert":{
  "Filtreringsdata":{
    "beskedtype":"MatrikulaerSagCreate",
    "beskedansvarligAktør":null,
    "tilladtModtager":null,
    "RelateretObjekt":null,
    "Objektregistrering":[
      {
        "registreringsaktør":"Anders Andersen",
        "registreringstid":null,
        "status":"Afsluttet",
        "objektansvarligAktør":"Geodatastyrelsen",
        "objektID":"ID20165",
        "objekttype":"MatrikulaerSag",
        "objekthandling":null,
        "opgaveemne":"52.20.05",
        "registreringsID":"5,0001-01-01T00:00:00.0000000",
        "Stedbestemmelse":null
      }
    ],
    "tværgåendeProces":"Diverse"
  },
  "Leveranceinformation":{
    "dannelsesetidspunkt":null,
    "transaktionsID":null,
    "kildesystem":null,
    "kildesystemIPAdresse":null,
    "kildesystemAkkreditiver":null,
    "sikkerhedsklassificering":"Ikke fortrolige data",
    "Leverancerute":null
  },
  "Modtagerhandling":null
},
"Beskeddata":[
  {
    "Objektreference":{"objektreference":"ID20165"},"
  }
]
Objektdata:null
}
}
}

```

```

--changesetresponse_f78a5a5e-53d4-42af-a334-ab7001cbe4fb
Content-Type: application/http
Content-Transfer-Encoding: binary
Content-ID: 2

```

```

HTTP/1.1 201 Created
Location: http:// host:1234/odata/Events(2)
Content-Type: application/json; odata.metadata=minimal;
charset=utf-8
OData-Version: 4.0

```

```

{
"@odata.context":"http://test.1234/odata/$metadata#Events
/$entity","Id":2,"Format":"JSON","Body":
  {"Hændelsesbesked":{
    "beskedversion":"1.0",
    "beskedId":null,
    "Beskedkuvert":{
      "Filtreringsdata":{
        "beskedtype":"MatrikulaerSagCreate",
        "beskedansvarligAktør":null,
        "tilladtModtager":null,
        "RelateretObjekt":null,
        "Objektregistrering":[
          {
            "registreringsaktør":"Anders Andersen",
            "registreringstid":null,
            "status":"Afsluttet",
            "objektansvarligAktør":"Geodatastyrelsen",
            "objektID":"ID20166",
            "objekttype":"MatrikulaerSag",
            "objekthandling":null,
            "opgaveemne":"52.20.05",
            "registreringsID":"5,0001-01-01T00:00:00.0000000",
            "Stedbestemmelse":null
          }
        ],
        "tværgåendeProces":"Diverse"
      },
      "Leveranceinformation":{
        "dannelsesetidspunkt":null,
        "transaktionsID":null,

```

```

        "kildesystem":null,
        "kildesystemIPAdresse":null,
        "kildesystemAkkreditiver":null,
        "sikkerhedsklassificering":"Ikke fortrolige data",
        "Leverancerute":null
    },
    "Modtagerhandling":null
},
"Beskeddata":[
    {"Objektreference":{"objektreference":"ID20166"},"
Objektdata":null}
]
    }}"
}
--changesetresponse_f78a5a5e-53d4-42af-a334-ab7001cbe4fb--
--batchresponse_f75bc63a-8b9e-4614-b081-c910c850d22c--

```

Ved oprettelse af en PUSH hændelse kan det være anbefales, at støtte sig til følgende:

Hændelses abonnement:

- Verificer at der er abonneret på de korrekte hændelser
- Verificer at de korrekte hændelsesabonnementer er aktive
- Verificer at abonnementets e-mail adresse er aktiv og anvendelig
- Verificer at det indtastede URL, der anvendes til at modtage hændelser, er korrekt. /\$batch er tilføjet til det konfigurerede URL af Datafordelerens system.

SSL terminator konfigurering:

- Kontroller client-auth SSL i det konfigurerede Datafordeler certifikats thumbprint
- Kontroller den interne port mapping i den applikation der modtager hændelser
- Kontroller den interne URL mapping i den applikation der modtager hændelser.

Testing:

- Test applikationen ved hjælp af en internet lokeret IP(ikke lokal), for at kontrollere Datafordeler EventPUSH klientens sti til netværket.
- Kontroller dit PUSH abonnementet ved at oprette et tilsvarende PULL abonnement.

I forbindelse med oprettelse af support sag, oplys følgende informationer:

- Navn på Webbruger/tjenestebruger
- Det URL som PUSH hændelsen forventes at være leveret på.
- Opsummering af problemet
- Applikationslogfiler i forbindelse med forbindelsen til applikationen (hvis tilgængeligt)
- Firewall/loadBalancer/Proxy/SSL terminerings logfiler for forbindelsen (hvis tilgængeligt)

En liste over de certifikater og udgående IP'er, der anvendes i forbindelse med PUSH hændelser af diverse Datafordeler miljøer, kan oplyses ved kontakt til Datafordeleren.

## Kodeeksempel til modtagelse af OData PUSH hændelser

Eksempel af brug findes her: [Modtag Data hændelsespakker fra datafordeleren](#)

## PULL af hændelser

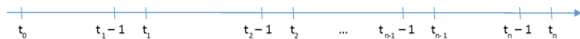
[Til toppen](#)

Anvenderen kan hente hændelsesbeskeder ved at kalde en tjeneste på Datafordeleren svarende til det opsatte abonnement. Ved PULL er det op til anvenderen at hente hændelsesbeskederne, som vil være tilgængelige umiddelbart efter, de er dannet. Hændelsesbeskederne vil være tilgængelige for afhentning i 2 måneder, hvorefter de slettes.

For en beskrivelse af, hvordan man opretter et abonnement på hændelser og henter hændelsesbeskeder, henviser vi til [Guide til hændelser på Datafordeleren](#) eller [Guide til hændelser på Selvbetjeningen](#).

Anvenderen kan tilgå tjenesten med enten brugernavn/password eller FOCES certifikat, afhængigt af hvilken sikkerhedszone tjenesten ligger i. Tjenesten leverer hændelser fra de abonnementer, som er tilknyttet den tjenestebruger, som man benytter til at tilgå tjenesten.

Hændelsesagenten registrerer nye hændelser i et commit-scope som tilføjes et timestamp. Laves der en forespørgelse samtidigt med at hændelserne er ved at blive skabt, risikerer man derved at miste en hændelse, da det igangværende scope tilføjes et timestamp der ligger før forespørgslen. For at sikre, at man ikke mister hændelser, anbefaler vi, at man laver sin forespørgelse med et forskudt tidsinterval på 1 minut før nu. Det vil sige man til tiden  $t_1$  forespørger på tidsintervallet fra  $t_0$  til  $t_1 - 1$  minut. Og derefter ved næste forespørgelse til tiden  $t_2$  forespørger på tidsintervallet fra  $t_1 - 1$  minut til  $t_2 - 1$  minut, osv. Til tiden  $t_n$  forespørger man således på tidsintervallet fra  $t_{n-1} - 1$  minut til  $t_n - 1$  minut.



Tjenesten fungerer som en REST-tjeneste på Datafordeleren.

Tjenesten findes på:

`<endpoint>/system/EventMessages/1.0.0/custom`

hvor `<endpoint>` afhænger af (i) hvilke hændelser, man abonnerer på, (ii) om man benytter brugernavn/adgangskode eller certifikat, og (iii) om man tilgår produktions- eller testmiljø. Der vil være et endpoint for adgang med kendt bruger med brugernavn/adgangskode, et endpoint for adgang med certifikat og et endpoint for adgang med certifikat til hændelsesbeskeder, der kan indeholde følsomme data.

Se [endpointoversigt og miljøer](#).

Et kald til tjenesten på et endpoint, hvor man kan benytte parametrene brugernavn /adgangskode for adgang med kendt bruger vil være på følgende form:

`<endpoint>/system/EventMessages/1.0.0/custom?datefrom=yyyy-mm-ddTHH:mm:ss&dateto=yyyy-mm-ddTHH:mm:ss&username=xxxx&password=yyyy`.

Datefrom og dateto samt brugernavn/adgangskode er obligatoriske parametre, hvilket er illustreret i følgende eksempel:

`https://services.datafordeler.dk/system/EventMessages/1.0.0/custom?datefrom=2020-07-03T00:00:00&dateto=2020-07-03T12:00:00&pagesize=100000&page=1&format=json&username=xxxxxx&password=yyyyyyy`

## Paging

PULL af hændelser kan højst levere 100.000 hændelser ad gangen. Store leverancer skal derfor opdeles i delleverancer.

Du kan administrere dine leverance ved at angive page og pagesize.

Ved leverancer med færre end 100.000 hændelser bliver samtlige hændelser levereret. Det er altid muligt at benytte page og pagesize til tilpasse størrelsen på leverancerne.

Brugseksempler:

1. <https://services.datafordeler.dk/system/EventMessages/1.0.0/custom?datefrom=2010-01-01&dateto=2017-12-22&format=Json>  
Returnerer page = 1, pagesize = 100.000.  
Du vil i dette eksempel modtage hændelse 0 til hændelse 100.000.
2. <https://services.datafordeler.dk/system/EventMessages/1.0.0/custom?datefrom=2010-01-01&dateto=2017-12-22&format=Json&page=1>  
Returnerer page = 1, pagesize = 100.000.  
Du vil i dette eksempel modtage hændelse 100.001 til hændelse 200.000.
3. <https://services.datafordeler.dk/system/EventMessages/1.0.0/custom?datefrom=2010-01-01&dateto=2017-12-22&format=Json&pagesize=10>  
Returnerer page = 1 pagesize = 10.

Du vil i dette eksempel modtage hændelse 0 til hændelse 10.

4. <https://services.datafordeler.dk/system/EventMessages/1.0.0/custom?datefrom=2010-01-01&dateto=2017-12-22&format=Json&pagesize=200000>  
Returnerer page = 1, pagesize = 100.000.  
Du vil i dette eksempel modtage hændelse 0 til hændelse 100.000.
5. <https://services.datafordeler.dk/system/EventMessages/1.0.0/custom?datefrom=2010-01-01&dateto=2017-12-22&format=Json&page=10&pagesize=20>  
Returnerer page = 10, pagesize = 20.  
Du vil i dette eksempel modtage hændelse 201 til hændelse 220.

---

## Kodeeksempel udstilling af hændelser på kort

Vil du bruge Datafordelerens kort via din egen webapplikation kan du hente inspiration fra [Datafordelerens kodeeksempler på GitHub](#)

På siden ligger [Kodeeksempel på hvordan hændelser fra Datafordeleren kan udstilles på et kort](#).

---

## Kvittering

Et af elementerne i beskedformatet er Modtagerhandling. Modtagerhandling elementet er valgfri og indeholder de valgfrie elementer Handling og Responsmodtager. I specifikationen for indholdet i beskedformatet, for en given beskedtype, kan Handling angives til Kvittering. Det betyder, at anvender kræves kvittering for alle hændelsesbeskedleverancer for den givne beskedtype. Anvenderen kvitterer for hver enkelt hændelsesbesked. Besked-identifikationen anvendes til at identificere hændelsesbeskeden over for kvitteringsservicen. Der benyttes samme kvitteringsfunktionalitet, som for kvitteringer for tjenester.

Kvitteringer gemmes i 30 dage, hvorefter de automatisk slettes. Sammen med kvitteringen gemmes besked-identifikationen og for datanære hændelser også objekt-identifikationen, da hændelsesbeskederne slettes løbende.

---

## Eksempel på hændelse fra opdatering til udsendelse

[Til toppen](#)

I dette afsnit vil vi gennemgå flowet fra opdatering til udsendelse af hændelser med et eksempel. Noget af eksemplet illustrerer, hvordan opdateringer fra et register styrer dannelsen af datanære hændelser. Det er en proces, som ikke er synlig for dataanvendere, men er taget med her af hensyn til fuldstændighed og introduktion til registre, som har behov for at forstå hele flowet. I eksemplet betragtes en simpel fejlrettelse – andre opdateringer kan give anledning til mere end én ny række i databasen. Bemærk at eksemplet kan afvige fra den endelige opsætning af det konkrete register på Datafordeleren, idet der er tale om et tænkt eksempel, der ikke nødvendigvis afspejler et forretningsmæssigt relevant scenarie.

Flowet starter ved at data bliver ændret i et register. Det kan fx være, at der er blevet konstateret en fejl i geometrien for en kommune og den bliver rettet – dvs. der sker en fejlrettelse i geometrien for en KommuneInddeling i DAGI-registreret, som ikke giver anledning til ændring i virkningstid. Denne opdatering af data bevirker, at der bliver sendt en opdatering til Datafordeleren.

Da der er bitemporalitet i data, bliver opdateringen foretaget ved, at den eksisterende forekomst af kommunen bliver gjort historisk, og der bliver oprettet en ny forekomst med den opdaterede geometri. Det sker i databasen ved at en eksisterende række får ændret RegistreringTil fra null til dags dato, og der bliver oprettet en ny række med RegistreringFra = dags dato, RegistreringTil = null og opdateret geometri i data.

Disse ændringer bliver sendt til Datafordeleren i XML med et element for oprettelsen og et element for ændringen (eksemplet er forkortet af hensyn til overskuelighed og kan afvige fra den endelige implementering):

#### Eksempel på besked fra registret til Datafordeleren

```
<?xml version='1.0' encoding='UTF-8'?>
<cmn:DatafordelerDataDelivery ... >
<cmn:Header>
...
</cmn:Header>
<cmn:Data>
  <cmn:DataEntity>
    <cmn:Action>Create</cmn:Action>
    <cmn:CreateEvent>UpdateEvent</cmn:CreateEvent>
    <dagi:Kommuneinddeling>
      <dagi:objectId>20059189</dagi:objectId>
      <dagi:feltliste>geometri</dagi:feltliste>
      ...
      <dagi:id_namespace>http://data.gov.dk/dagi</dagi:
id_namespace>
      <dagi:id_lokalId>10389119</dagi:id_lokalId>
      ...
      <dagi:landekode>DK</dagi:landekode>
      <dagi:skala>1:10.000</dagi:skala>
      <dagi:geometri> ... </dagi:geometri>
      ...
    </dagi:Kommuneinddeling>
  </cmn:DataEntity>
  <cmn:DataEntity>
    <cmn:Action>Update</cmn:Action>
    <cmn:CreateEvent>NoEvent</cmn:CreateEvent>
    <dagi:Kommuneinddeling>
      <dagi:objectId>10059189</dagi:objectId>
      <dagi:registreringTil>2016-08-07T02:10:00.000000</dagi:
registreringTil>
    </dagi:Kommuneinddeling>
  </cmn:DataEntity>
</cmn:Data>
</cmn:DatafordelerDataDelivery>
```

Den eksisterende række bliver opdateret og gjort historisk (man siger også, at rækken bliver afregistreret) – dette skal ikke udløse en hændelse. Oprettelsen af en ny række med opdateret data bliver derimod brugt til at trigge dannelse af en datanær hændelse ved at angive elementet <CreateEvent> med værdien UpdateEvent. Da der set fra et forretningsmæssigt perspektiv er tale om en opdatering af data, dannes der en Update-hændelse.

Dette er et eksempel på, at typen af den datanære hændelse ikke er bestemt af databaseoperationen, men at det alene er registeret, der styrer beskedtypen ved et element i opdateringen.

Der dannes en datanær hændelsesbesked med beskedtypen KommuneInddelingUpdate. Det er i forvejen konfigureret, hvilke informationer, der skal fyldes i en hændelsesbesked af denne type. Informationerne hentes typisk fra dataopdateringen selv.

Dataopdateringer kan indeholde oplysninger, som ikke er en del af de data, der gemmes og udstilles af datafordeleren, men som udelukkende benyttes til at danne hændelsesbeskeder. Elementet feltliste i dataopdateringen indeholder en liste af navne på felter, som er ændret ved dataopdateringen. Ved ændring af geometrien for en kommune, indeholder opdateringen "geometri" i elementet feltliste.

Som anvender skal man i forvejen have sat et abonnement op for at få den hændelsesbesked, der bliver dannet. Hvis man i sit abonnement har valgt at hente hændelsesbeskeder (PULL-scenariet), vil hændelsesbeskeden blive gjort tilgængelig i en RESTful tjeneste med følgende signatur:

<https://services.datafordeler.dk/system/EventMessages/1.0.0/custom?datefrom=2016-10-06&dateto=2016-10-07&format=json&username=xxx&password=yyy>

Ved at kalde tjenesten, kan man hente hændelsesbeskederne for en given periode for de abonnenter, der er sat op for anvenderen. Det er først her, at anvenderen kommer i kontakt med hændelserne – dataopdateringerne fra registeret til Datafordeleren er ikke direkte synlige for anvenderen, men bliver afspejlet i de genererede hændelser.

En hændelsesbesked dannet ud fra opdateringen i eksemplet kunne se ud som følger (eksemplet er forkortet af hensyn til overskuelighed og kan afvige fra den endelige implementering):

#### Eksempel på besked fra Datafordeleren til en anvender

```
<?xml version="1.0" encoding="utf-16"?>
<Haendelsesbesked xmlns:urn="urn:dk:grunddata:1.0.0" xmlns="urn">
  <Beskedversion>1.0</Beskedversion>
  <BeskedId>f90c19c0-39da-45f2-9093-fd6afe4335ca</BeskedId>
  <Beskedkuvert>
    <Filtreringsdata>
      <Beskedtype>KommuneinddelingUpdate</Beskedtype>
      <BeskedansvarligAktoer />
      <RelateretObjekt />
      <Objektregistrering>
        <Registreringsaktoer>Systemetablering<
/Registreringsaktoer>
        <Registreringstid>2016-08-07T02:10:00.000000<
/Registreringstid>
        <Status>vedtaget</Status>
        <ObjektansvarligAktoer>Social- og
Indenrigsministeriet</ObjektansvarligAktoer>
        <ObjektId>20059189</ObjektId>
        <Objekttype>Kommuneinddeling</Objekttype>
        <Objekthandling>geometri</Objekthandling>
        <Opgaveemne>52.20.10.15</Opgaveemne>
        <RegistreringsId>10389119, 2016-08-07T02:10:00.000000<
/RegistreringsId>
        <Stedbestemmelse>
          <StedbestemmelseReference>MULTIPOLYGON
(((738763.98 6169447.66 ... )))
          </StedbestemmelseReference>
        </Stedbestemmelse>
      </Objektregistrering>
      <TvaergaaendeProces>systemetablering<
/TvaergaaendeProces>
    </Filtreringsdata>
    <Leveranceinformation>
      <Dannelsestidspunkt>2016-08-07T07:01:24.5270830+02:00<
/Dannelsestidspunkt>
      <Kildesystem>http://data.gov.dk/id/itsystem
/basicdata#d45f25737a3972bd68
ec2d9f043bb7e479fc1b6f</Kildesystem>
      <Sikkerhedsklassificering>http://data.gov.dk/vocabulary
/security/ confidentiality#NonConfidential<
/Sikkerhedsklassificering>
    </Leveranceinformation>
  </Beskedkuvert>
  <Beskeddata>
    <Objektreference>20059189</Objektreference>
  </Beskeddata>
</Haendelsesbesked>
```

Når hændelsesbeskeden hentes (PULL-scenariet), vil den være pakket ind i et format, der kan indeholde flere hændelsesbeskeder:

### Eksempel på besked fra Datafordeleren til en anvender med flere

#### hændelsesbeskeder

```
<?xml version="1.0" encoding="utf-8"?>
<ArrayOfEnvelope xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Envelope>
    <Id>1878</Id>
    <Message>
      <?xml version="1.0" encoding="utf-16"?>
      <Haendelsesbesked xmlns:urn="urn:dk:grunddata:1.0.0" xmlns="urn">
        <Beskedversion>1.0</Beskedversion>
        <BeskedId>f90c19c0-39da-45f2-9093-fd6afe4335ca</BeskedId>
        ...
      </Haendelsesbesked>
    </Message>
    <Format>Xml</Format>
    <Timestamp>2016-08-07T07:02:17.0601880+02:00</Timestamp>
  </Envelope>
</ArrayOfEnvelope>
```

Når hændelsesbeskeden sendes til anvenderen (PUSH-scenariet), vil den tilsvarende være pakket ind i et JSON-format.

[Til toppen](#)

## Anvendelse af hændelser

Hændelser benyttes til at binde processer på tværs af registrene sammen. Når en forvaltningsproces opdaterer data i ét register, kan det igangsætte en forvaltningsproces for opdatering af data i et andet register.

## Brugscenarier

Hvis du selv har data, som er relateret til grunddata eller afledt af grunddata, vil de løbende skulle opdateres som konsekvens af ændringer i grunddata. Her kan hændelser benyttes til at trigge opdatering af dine data.

Et simpelt eksempel på data afledt af grunddata kunne være et såkaldt heatmap for matrikulære aktiviteter. Det vil sige et kort, hvor farven for et område afhænger af antallet af udstykninger inden for en given periode, f.eks. det seneste år. Dette kort kan genereres ud fra matrikulære data med historiske oplysninger. I stedet for at generere kortet gentagne gange, kunne man vedligeholde datagrundlaget for kortet ud fra ændringer, som der løbende kommer hændelser om.

Det kan også være, at du har behov for at kende til begivenheder af hensyn til din forretning. Det kan f.eks. være, at en finansiel part modtager en hændelse om ejerskifte for en bestemt fast ejendom, som vedkommende har pant i. Et andet eksempel kan være, at en offentlig myndighed eller privat aktør, der administrerer ejendomme, modtager en hændelse om, at en bestemt fast ejendom har fået ny administrator. Det vil sige hændelser kan trigge aktiviteter i forretningsprocesser hos forskellige parter.

## Hændelser versus deltafiler

En del anvendere har i dag lokale kopier af data liggende i deres systemer og opdaterer disse "skyggeregistre" ved hjælp af såkaldte deltafiler, det vil sige filer med ændringer i data. Hændelser indeholder oplysninger om, at data er opdateret, og kan derfor i princippet benyttes til at trigge opdatering eller vedligehold af lokale data hos en dataanvender. Mulighederne afhænger dog af, hvilke hændelser, det enkelte register har valgt at udstille.

Datafordeleren danner ikke delinformation i forbindelse med datanære hændelser. Som tidligere omtalt, vil nogle registre sende oplysninger om opdaterede felter i feltet Objekthandling og der kan medsendes oplysninger i beskedsdata, men det vil som udgangspunkt være op til at anvenderen at hente tidligere og nye versioner af objekter, hvis der er behov for at sammenligne. I hændelsesbeskederne angives typisk referencer til objekter i form af deres id'er.

Data med bitemporale egenskaber kan udstilles i tjenester på en sådan måde, at man kan fremsøge data for bestemte tidspunkter og tidsperioder. Med sådanne tjenester vil det derfor være muligt at fremsøge historiske data, således at man kan få historik for data uden at have abonneret på hændelser i en given periode. Registrerne bestemmer hvilke tjenester, der skal udbydes på Datafordeleren, og det er således op til registrerne at understøtte behov for historik i tjenester og eventuelle særlige tjenester.

---

## Tidsdimensioner

Når man anvender hændelser, skal man være opmærksom på, at datanære hændelser bliver genereret og distribueret ved opdatering af Datafordeleren. Det sker, når der er sket en registrering, og hændelser foregår derfor i relation til registreringstid. En registrering kan have en virkningstid, der ligger ude i fremtiden, men hændelsen kommer altså ved registreringen (helt præcist ved den efterfølgende opdatering til Datafordeleren) og ikke som følge af, at virkningen træder i kraft (virkningstiden). Hvis man har behov for at reagere i forhold til virkningstiden, skal man selv holde styr på ændringerne i forhold til den.

Hvis man f.eks. registrerer en ændring den 4. januar 2017, som har virkning fra den 1. februar 2017, så vil der blive dannet en datanær hændelse om ændringen den 4. januar 2017, men der vil ikke blive dannet nogen hændelse den 1. februar 2017 om at ændringen nu har virkning.

---

## Andre abonnementer

[Til toppen](#)

Der er andre former for abonnementer end hændelsesabonnementer på Datafordeleren, og det kan give anledning til forvirring, når der blot tales om "abonnementer". Det er muligt at opsætte abonnementer på filudtræk og at abonnere på opdateringer af tjenester ved hjælp af Atom feeds.

---

## Abonnementer på filudtræk

Anvendere kan oprette abonnement på et filudtræk, der udvælger en delmængde af data og dannes ved et angivet tidsinterval. Når der er genereret et nyt filudtræk (med nye data), får anvenderen en e-mail med link til filudtrækket. Filudtrækket kan hentes via FTP, SFTP eller en tjeneste på Datafordeleren, alt efter hvordan man identificerer sig (brugernavn/adgangskode, certifikater eller SSH2-nøgler) og sikkerhedsniveau for data.

Læs mere på [Filudtræk på Datafordeleren](#).

---

## Abonnement på Atom feeds

Anvendere har mulighed for at blive adviseret, når en ny version af en given tjeneste publiceres. I Selvbetjeningen findes oplysninger om Atom feeds for tjenester under detaljevisningen – her kan man finde og kopiere webadressen på tjenestens Atom feed. Man kan herefter oprette et feed-abonnement ved fx at tilføje et nyt feed i Microsoft Outlook eller en anden applikation med angivelse af webadressen på tjenestens Atom feed.

Når man har sat et feed-abonnement op for en given tjeneste, bliver man adviseret, når en ny version af tjenesten publiceres. Det sker ved, at den applikation (feed reader), man benytter, ved jævne mellemrum tjekker for nyt indhold i Atom feed'et på Datafordeleren.

