

GraphQL eksempler mod Datafordeleren

Denne guide viser en række praktiske eksempler på, hvordan man bygger GraphQL forespørgsler mod datafordeleren. Det antages i denne guide, at du har din egen API-key til Datafordeleren. Denne skal indsættes i alle eksempler, hvor der er anført **API-KEY**

Forfatter: Morten Fuglsang, Septima

Hentning af schemaer

Schemaer kan kaldes med en GET kommando, og returnerer en graphql schema-fil hvis den kaldes i en browser. Denne kan med fordel indlæses i diverse editors.

DAR

<https://graphql.datafordeler.dk/DAR/v1/schema?apiKey=API-KEY>

BBR

<https://graphql.datafordeler.dk/BBR/v1/schema?apiKey=API-KEY>

DAGI

<https://graphql.datafordeler.dk/DAGI/v1/schema?apiKey=API-KEY>

Stednavne

<https://graphql.datafordeler.dk/DS/v1/schema?apiKey=API-KEY>

CVR

<https://graphql.datafordeler.dk/CVR/v1/schema?apiKey=API-KEY>

Ejendomsbeliggenhedsregistret

<https://graphql.datafordeler.dk/EBR/v1/schema?apiKey=API-KEY>

Ejerfortegnelsen

<https://graphql.datafordeler.dk/EJF/v1/schema?apiKey=API-KEY>

GeoDanmark Vektor

<https://graphql.datafordeler.dk/GEODKV/v1/schema?apiKey=API-KEY>

Matriklen2

<https://graphql.datafordeler.dk/MAT/v1/schema?apiKey=API-KEY>

Basis forespørgsler

DAR

Endpoint : <https://graphql.datafordeler.dk/DAR/v1?apiKey=API-KEY>

```
query {
  DAR_Adresse(
    virkningstid: "2025-05-04T00:00:00Z"
    first: 10
  ) {
    nodes {
      adressebetegnelse
      id_lokalId
      husnummer
      doerbetegnelse
      etagebetegnelse
      status
      registreringFra
      virkningFra
    }
  }
}
```

```
query {
  DAR_Husnummer(first: 10, virkningstid: "2025-05-04T00:00:00Z") {
    nodes {
      adgangsadressebetegnelse
      husnummertekst
      id_lokalId
      postnummer
      status
      registreringFra
      virkningFra
    }
  }
}
```

```
query {
  DAR_Adressepunkt(first: 10, virkningstid: "2025-05-04T00:00:00Z") {
    nodes {
      id_lokalId
      status
      oprindelse_kilde
      oprindelse_noejagtighedsklasse
      oprindelse_tekniskStandard
      position {
        wkt
        crs
        type
      }
    }
  }
}
```

```

        dimension
      }
      registreringFra
      virkningFra
    }
  }
}

```

```

query {
  DAR_NavngivenVej(first: 10, virkningstid: "2025-06-04T00:00:00Z") {
    nodes {
      id_lokalId
      vejnavn
      vejadresseringsnavn
      udtaltVejnavn
      administreresAfKommune
      status
      registreringFra
      virkningFra
    }
  }
}

```

Stednavne

Endpoint <https://graphql.datafordeler.dk/DS/v1?apiKey=API-KEY>

```

query {
  DS_Bebyggelse(
    first: 10
    virkningstid: "2025-06-04T00:00:00Z"
  ) {
    nodes {
      id_lokalId
      bebyggelseskode
      bebyggelsestype
      areal
      indbyggertal
      geometri {
        wkt
        crs
      }
    }
  }
}

```

Matriklen

Endpoint : <https://graphql.datafordeler.dk/MAT/v1?apiKey=API-KEY>

```
query {
  MAT_Jordstykke(
    first: 10
    virkningstid: "2025-06-04T00:00:00Z"
  ) {
    nodes {
      id_lokalId
      matrikelnummer
      delnummer
      ejerlavLokalId
      kommuneLokalId
      regionLokalId
      sognLokalId
      status
      registreretAreal
      arealberegningstype
      arealbetegnelse
      arealtype
      brugsretsareal
      faelleslod
      fredskov_areal
      fredskov_omfang
      jordrente_omfang
      klitfredning_areal
      klitfredning_omfang
      majoratsskov_nummer
      majoratsskov_omfang
      strandbeskyttelse_areal
      strandbeskyttelse_omfang
      vejareal
      vejarealberegningstype
      vandarealinkludering
      samletFastEjendomLokalId
      senesteSagLokalId
      skelforretningssagsLokalId
      supplerendeMaalingSagLokalId
    }
  }
}
```

```
query {
  MAT_SamletFastEjendom(
    first: 10
    virkningstid: "2025-06-04T00:00:00Z"
  ) {
    nodes {
      id_lokalId
      id_namespace
    }
  }
}
```



```

    administreresAfKommune
  }
}
}

```

```

query {
  DAR_Adresse(
    where: {
      adressebetegnelse: { eq: "Lundagervej 43, 2740 Skovlunde" }
    }
    first: 1
    virkningstid: "2025-06-04T00:00:00Z"
  ) {
    nodes {
      adressebetegnelse
      husnummer
      doerbetegnelse
      etagebetegnelse
      status
    }
  }
}

```

Endpoint <https://graphql.datafordeler.dk/MAT/v1?apiKey=API-KEY>

```

query {
  MAT_Jordstykke(
    where: {
      matrikelnummer: { eq: "6hh" }
      ejerlavLokalId: { eq: "21751" }
    }
    first: 1
    virkningstid: "2025-06-04T00:00:00"
  ) {
    nodes {
      id_lokalId
      matrikelnummer
      ejerlavLokalId
      registreretAreal
      status
    }
  }
}

```

```

query {
  MAT_SamletFastEjendom(
    where: {

```

```

        BFEnummer: { eq: 2154708 }
      }
      first: 1
      virkningstid: "2025-06-04T00:00:00"
    ) {
      nodes {
        id_lokalId
        BFEnummer
        status
        arbejderbolig
        erFaelleslod
        landbrugsnotering
        udskiltVej
        hovedejendomOpdeltIEjerlejligh
        geometri {
          wkt
          crs
        }
      }
    }
  }
}

```

IN Filter

GraphQL understøtter 'IN LIST' filtre - her er eksemplet fra stednavne :

Endpoint <https://graphql.datafordeler.dk/DS/v1?apiKey=API-KEY>

```

query {
  DS_Bebyggelse(
    where: {
      bebyggelsestype: { in: ["by", "bydel"] }
    }
    first: 10
    virkningstid: "2025-06-04T00:00:00Z"
  ) {
    nodes {
      id_lokalId
      bebyggelsestype
      bebyggelseskode
      indbyggertal
      geometri {
        wkt
        crs
      }
    }
  }
}

```

Forespørgsler med paging

Paging gør det muligt at hente resultater i chunks - så man ikke skal vente på meget store forespørgsler returnerer alt på en gang.

```
query {
  DAR_Adresse(
    first: 10
    virkningstid: "2025-06-04T00:00:00Z"
  ) {
    pageInfo {
      endCursor
      hasNextPage
    }
    nodes {
      id_lokalId
      adressebetegnelse
    }
  }
}
```

Dette svar giver en cursor tilbage, der gør at vi kan hente de næste 10 ved at bruge denne cursor.

```
  "pageInfo": {
    "endCursor":
    "TURBd01EWTNNR010TkDZNE9TMDBZekEzTFdJM04ySXRaamxpT0RKaFpqQXhZemd3001qQX1
    NQzB3TnkweE4xUXhNRG94T1RveE1pNDJOakV6TmpFd0t6QXdPakF3001qQX1NQzB3TnkweE4
    xUXhNRG94T1RveE1pNDJOakV6TmpFd0t6QXdPakF3",
    "hasNextPage": true
  },
```

```
query {
  DAR_Adresse(
    first: 10
    after:
    "TURBd01EWTNNR010TkDZNE9TMDBZekEzTFdJM04ySXRaamxpT0RKaFpqQXhZemd3001qQX1
    NQzB3TnkweE4xUXhNRG94T1RveE1pNDJOakV6TmpFd0t6QXdPakF3001qQX1NQzB3TnkweE4
    xUXhNRG94T1RveE1pNDJOakV6TmpFd0t6QXdPakF3",
    virkningstid: "2025-06-04T00:00:00Z"
  ) {
    pageInfo {
      endCursor
      hasNextPage
    }
    nodes {
      id_lokalId
      adressebetegnelse
    }
  }
}
```


Dette kan foresættes indtil

```
"hasNextPage": true
```

bliver false - så har vi hentet alle adresser for det givne tidspunkt - 10 ad gangen...

Spatiale forespørgsler

For at kunne lave en spatial forespørgsel, skal data indeholde geometri. Derefter skal det undersøges, hvilke spatiale operatører der er til rådighed fra schemaet.

```
query {
  DAR_Adressepunkt(
    where: {
      position: {
        within: {
          wkt: "POLYGON((712994.1671187154
6179882.641955254,713134.5416895023 6179882.641955254,713134.5416895023
6179992.936260872,712994.1671187154 6179992.936260872,712994.1671187154
6179882.641955254))"
          crs: 25832
        }
      }
    }
    first: 100
    virkningstid: "2025-06-04T00:00:00Z"
  ) {
    nodes {
      id_lokalId
      status
      position {
        wkt
      }
    }
  }
}
```

Matriklen

Samlet fast ejendom ud fra BFE-nummer indenfor en søge-geometri

Endpoint <https://graphql.datafordeler.dk/MAT/v1?apiKey=API-KEY>

```
query {
  MAT_SamletFastEjendom(
    where: {
```

```

      geometri: {
        within: {
          wkt: "POLYGON((712994.1671187154
6179882.641955254,713134.5416895023 6179882.641955254,713134.5416895023
6179992.936260872,712994.1671187154 6179992.936260872,712994.1671187154
6179882.641955254))"
          crs: 25832
        }
      }
    }
    first: 100
    virkningstid: "2025-06-04T00:00:00Z"
  ) {
    nodes {
      BFEnummer
      id_lokalId
      geometri {
        wkt
      }
    }
  }
}

```

Bitemporale forespørgsler

Forespørgsler med relationer

Stednavne

Hvis vi i stednavne vil have både navn og geometri, så skal vi kombinere to forespørgsler: Først henter vi de stednavne vi gerne vil finde. Når vi har fundet dem, så taget vi navngivetSted_objectid ud

```

query {
  DS_Stednavn(
    where: {
      skrivemaade: { eq: "Skovlunde" }
    }
    first: 10
  ) {
    nodes {
      skrivemaade
      navngivetSted_objectid
    }
  }
}

```

Herefter kan vi så spørge efter geometrien dertil ud fra navngivetSted_objectid:

```
query {  
  DS_Bebyggelse(  
    where: {  
      objectid: { eq: "476543" }  
    }  
    virkningstid: "2025-06-04T00:00:00"  
  ) {  
    nodes {  
      id_lokalId  
      objectid  
      bebyggelsestype  
      indbyggertal  
      geometri {  
        wkt  
      }  
    }  
  }  
}
```

Forespørgsler på tværs af registre