

# REST på Datafordeleren

Oprindelig kilde	Datafordeleren
Forfatter	Datafordeleren
Oprettet	Jun 28, 2019
Version	1.0
Ændret	Oct 30, 2023
Sidehistorik	<ul style="list-style-type: none"><li>Se sidehistorik</li></ul>

Siden giver en introduktion til og vejledning i, hvordan man kan anvende webservices som udstilles på Datafordeleren.

Gå til [Guide til REST i Selvbetjeningen](#), hvis du har brug for helt konkrete anvisninger til at bruge en webservice fra Datafordeleren.

I de registerspecifikke servicekataloger vil du under hver metode kunne se eksempler på hvordan url'en er opbygget og på siderne [Datafordeleren - eksempel på anvendelse af bitemporalitet](#) og [User stories](#) kan du se eksempler på hvordan url'er bliver opbygget og brugt.

- Fordele ved brug af REST
- Servicetyper
- Opbygning af url for en webservice
- Parametre
  - Sikkerhedsparametre
    - OCES certifikater
    - Brugernavn /adgangskode
  - Standardparametre
    - Paging
    - Count
    - Format
    - jsonpCallback
  - Tjenestespecifikke parametre
    - Liste af værdier
    - Date og DateTime
    - Obligatoriske parametre
    - Ikke-obligatoriske parametre
- Kvitteringer
- Token
- Statuskoder

### BEMÆRK

På Datafordeleren er der ikke konfigureret timeout på webtjenester.

## Fordele ved brug af REST

REST-tjenester i princippet er designet til relativt simple ad-hoc forespørgsler, der returnerer begrænsede datamængder.

[Til toppen](#)

Med REST-tjenesterne finder du eksempelvis svar på et ønsket øjebliksbillede som:

- Hvilke registreringer har aktuel virkning?
- Hvilke registreringer havde virkning på tidspunkt x?
- Hvornår blev en given registrering foretaget vedrørende objekt A med virkning på tidspunkt x?

---

## Service typer

Webservices giver anvenderne mulighed for at hente data ved at kalde en service.

[Til toppen](#)

Datafordeleren understøtter følgende servicetyper:

- **WMS** Web Mapping Service
- **WFS** Web Feature Service
- **WCS** Web Coverage Service
- **WMTS** Web Map Tile Service
- **REST** REpresentational State Transfer

De følgende afsnit vil primært beskrive REST-services.

---

## Opbygning af url for en webservice

Når en webservice konfigureres genereres der automatisk en url.

[Til toppen](#)

Opbygningen af url'en sker ud fra følgende kriterier, som angives i forbindelse med konfigurationen af webservicen:

- Protokol (http/https)
- Webservicetype (REST, WFS, WMS, WMTS, WCS)
- Webservicenavn
- Version af webservicen

Strukturen for den genererede url sker i overensstemmelse med følgende skema:

```
<scheme>://<authority>/<path>/?<parameters>
```

Strukturen for den samlede url sker i overensstemmelse med følgende skema, når <authority> og <path> bliver foldet ud:

```
<scheme>://<alias>.datafordeler.dk/<owner>/<service>  
/<version>/<metode>/?<parameters>
```

## BESKRIVELSE AF URL-OPBYGNING

**<scheme>** afgøres af den valgte protokol og vil altid være enten http eller https.

Det anbefales ikke at sende brugernavn/adgangskode ukrypteret. Derfor bør der altid anvendes https, hvis disse sikkerhedsakkreditiver anvendes.

**<authority>** udgøres af følgende mønster **<alias>**.  
[datafordeler.dk](#)

- **<alias>**  
Når webservicen konfigureres op i Datafordelerens testmiljø, vil **<alias>** blive valgt ud fra en pulje af tilgængelige aliaser, som er tilknyttet den i konfigurationen valgte webservicetype.

**<path>**udgøres af følgende mønster **<owner>/<service>/<version>/<metode>**

- **<owner>**  
Angiver hvilket register der ejer webservicen og sikrer at der tillades navnesammenfald på tjenester på tværs af registre, men ikke inden for det samme register.
- **<service>**  
Angiver webservicens navn.
- **<version>**  
Angiver hvilken version af webservicen der er tale om.
- **<metode>**  
Angives afhængigt af webservicetypen. Ved DAGI-tjenester kan der angives WFS eller WMS. For REST-service angives der REST samt eventuelt navn på metoden fx REST/adresse, da en REST-service kan have flere metoder.

**<parameters>**  
I slutningen af url'en angives parametre, som bliver beskrevet nærmere i det efterfølgende afsnit.

---

## Parametre

Parametre dækker over både sikkerhedsparmetre, standardparametre og tjenestespecifikke parametre.

[Til toppen](#)

---

## Sikkerhedsparmetre

Datafordeleren understøtter en række forskellige akkreditiver. De skal sendes med i kald til tjenester, der ikke har anonym adgang.

Brug [Dataoversigten](#) eller de registerspecifikke servicekataloger til at undersøge, hvilken brugeradgang den ønskede tjeneste kræver og læs mere om [Brugeroprettelse](#) og [Brugeradgang](#) på [Datafordeler.dk](#).

---

## OCES certifikater

Kald med certifikater skal ikke ske til det "normale" endpoint, men skal gå til et separat endpoint specifikt for anvendelse af certifikater.

- Url vil for webservices med certifikater i zone 0 starte med: <https://certservices.datafordeler.dk/>
- Url vil for webservices med certifikater i zone 5 starte med: <https://s5-certservices.datafordeler.dk/>

Se endpoint for testmiljøerne på siden [Hent data](#).

---

## Brugernavn/adgangskode

Vil du benytte brugernavn/adgangskode som akkreditiver, skal parametre inkluderes i kald til tjeneste og den samlede url vil følge dette skema:

```
<scheme>://<alias>.datafordeler.dk/<owner>/<service>  
/<version>/<metode>/?username=xxx&password=yyy
```

### INDSÆT OPLYSNINGER I URL'EN FOR BRUGERNAVN /ADGANGSKODE

Brugernavn og adgangskode er obligatoriske parametre, hvis tjenestebrugeren er oprettet på den måde. Det er tjenestebrugers brugernavn og adgangskode, som skal tilføjes. Følgende skal tilføjes url'en for at hente hændelsesbeskeder:

xxx - skal erstattes med brugernavn for tjenestebruger  
yyy - skal erstattes med adgangskode for tjenestebruger

Nogle tjenester vil du med andre ord kunne tilgå med både brugernavn/adgangskode og certifikat. Andre tjenester vil du af sikkerhedsmæssige årsager kun kunne tilgå med certifikat.

I de registerspecifikke servicekataloger vil du under hver metode kunne se eksempler på hvordan url'en er opbygget og hvordan du kan tilgå den.

---

## Standardparametre

En række parametre er standard på Datafordeleren. I webservicekatalogerne vil det fremgå om registrets webservices er konfigureret til at benytte disse parametre.

---

## Paging

Det er muligt at anvende paging for de REST-services, der er konfigureret til det.

I webservicekatalogerne vil det fremgå af de generelle informationer om registrets webservices er konfigureret til at benytte paging samt informationer om en default paging-størrelse.

Anvender paging:

- Bygnings- og Boligregistret (BBR) (Default paging-størrelse: 100)
- Danmarks Administrative Geografiske Inddeling (DAGI) (Default paging-størrelse: 20)
- Danmarks Adresseregister (DAR) (Default paging-størrelse: 100)
- Det Centrale Personregister (CPR) - offentlig (Default paging-størrelse: 20)
- Ejendomsbeliggenhedsregistret (EBR) (Default paging-størrelse: 20)
- Ejerfortegnelsen (EJF) (Default paging-størrelse: 20)

Anvender ikke paging:

- Danmarks Højdemodel (DHM)
- Danske Stednavne
- Det Centrale Personregister (CPR) - privat
- Det Centrale Virksomhedsregister (CVR)
- Ejendomsvurdering (VUR)
- Historiske kort og data
- Matriklen (MAT)

Benytter du paging vil den samlede url følge dette skema:

```
<scheme>://<alias>.datafordeler.dk/<owner>/<service>
/<version>/<metode>/?page=xxx&pagesize=yyy
```

#### INDSÆT OPLYSNINGER I URL'EN FOR PAGING

Følgende skal tilføjes url'en for at hente hændelsesbeskeder med paging:

xxx - skal erstattes med nummeret på den side, som du ønsker.  
yyy - skal erstattes med den sidestørrelse, som du ønsker.

## Count

Antal elementer i resultatsættet returneres ved at angive parameteren count i REST kald.

Benytter du count vil den samlede url følge dette skema:

```
<scheme>://<alias>.datafordeler.dk/<owner>/<service>
/<version>/<metode>/?count=true
```

#### INDSÆT OPLYSNINGER I URL'EN FOR COUNT

Følgende skal tilføjes url'en for at hente antallet af elementer i resultatsættet:

count=true

## Format

REST-servicen kan returnere data i formaterne JSON og XML. An giver du ikke hvilket format du ønsker dit output vil du modtage det i JSON, da de er defaultværdien på Datafordeleren.

Vælger du at specificere formatet, vil den samlede url følge dette skema:

```
<scheme>://<alias>.datafordeler.dk/<owner>/<service>
/<version>/<metode>/?format=xxx
```

#### INDSÆT OPLYSNINGER I URL'EN FOR FORMAT

Følgende skal tilføjes url'en for at hente hændelsesbeskeder med et bestemt format:

xxx - skal erstattes med enten json eller xml.

En anden mulighed er at sætte accept headeren til "application /json" eller "application/xml".

---

## jsonpCallback

jsonpCallback: jsonp kan anvendes ved at angive jsonpCallback=xxx.

---

## Tjenestespecifikke parametre

De registerspecifikke servicekataloger er knyttet til de specifikationer, som registret har specificeret i deres Dataleverancespecifikation (DLS). For hver parameter er dens type angivet.

Der benyttes følgende typer:

- UUID
- Liste af værdier af typen UUID
- String
- Liste af værdier af typen String
- Integer
- Liste af værdier af typen Integer
- DateTime
- Date
- Boolean
- Double

I de registerspecifikke servicekataloger vil du under hver metode kunne se hvilke parametre, som det er muligt at benytte på den enkelt metode og af hvilken type den er.

---

## Liste af værdier

Nogle tjenester accepterer at en parameter består af en liste af værdier.

For de REST baserede webservices er dette understøttet ved at de enkelte værdier i listen adskilles af et "|" -tegn (pipe karakter).

Eksempel med 4 værdier i en liste til en parameter: V  
ærdi1|Værdi2|Værdi3|Værdi4

#### EKSEMPEL PÅ KALD MED FLERE CPR NUMRE

```
https://s5-certservices.datafordeler.dk/CPR  
/CprPersonFullComplete/1/rest/PersonFullListComplete?  
Format=JSON&pnr.personnummer.  
wi=xxxxxxxxxx|xxxxxxxxxx
```

---

## Date og DateTime

Datoformat følger ISO 8601 Date and time format. Alle datoer skrives med andre ord i formatet yyyy-mm-ddTHH:MM:SS.sssss, hvor tidsangivelsen kan udelades.

Tidszonen kan angives enten ved (2019-02-05T07:50:14.687580+01:00 (for normaltid /vintertid)), (2019-07-05T07:50:14.687580+02:00 (for sommertid)) eller ved at benytte Z (2019-07-05T07:50:14.687580Z).

---

## Obligatoriske parametre

For at kalde en stor del af metoderne på Datafordeleren er det nødvendigt at tilføje obligatoriske parametre.

I registerspecifikke servicekatalogerne vil du under hver metode se eksempler på hvordan url'en er opbygget og i disse eksempler er tilføjet eksempler på de obligatoriske parametre.

Hvordan du skal skrive de obligatoriske parametre er ikke ensrettet på tværs af registre og det er derfor vigtigt, at du opbygger url'en på baggrund af oplysningerne i servicekataloget for den metode, som du er interesseret i at benytte.

---

## Ikke-obligatoriske parametre

For hver ikke-obligatorisk parameter er der en default værdi. Hvis parameteren udelades i et kald til en tjeneste, så benytter tjenesten default værdien for parameteren ved behandling af kaldet.

I de registerspecifikke servicekataloger er angivet default værdier for hvert enkelt parameter.

---

## Kvitteringer

Webservices kan i Datafordeleren konfigureres til at kræve kvitteringer fra anvenderen.

[Til toppen](#)

Alle forespørgsler til en webservice vil returnere en response-header med kvitteringsID (receipt id) og kaldet DataDistributor. ReceiptId.

For at kvittere for data skal du sende en POST anmodning til kvitteringsservicen.

<https://<endpoint>.datafordeler.dk/system/receipt/1/rest?username=xxx&password=yyy>

med en request body der ser ud som (json eksempel):

```
{ "ReceiptId": "zzz" }
```

hvor "zzz" angiver kvitteringsID.

---

## Token

Det skal bemærkes, at dette afsnit beskriver anvendelse af Datafordelerens egen Token Server, og det ikke er den fællesoffentlige Token service fra NemLog-in (Nemlog-in STS - Security Token Service).

[Til toppen](#)

For at benytte SAML2 autentifikation skal du anskaffe en SAML2 token fra Token serveren og tilføje denne token til autentifikationsheaderen i dit kald.

Følgende eksempel viser hvordan det er gjort i C#.NET, hvor tjenestebrugerens brugernavn er erstattet med xxx og adgangskode med yyy.



### EKSEMPEL PÅ TOKEN I C#.NET

```
string ADFS_URL = "https://fs.datafordeler.dk/adfs/services/trust/13/usernamemixed";
string REALM = "urn:dk:kmd:dd:value:sp:switchboard:knownuser";
string USER_NAME = "dfdprod01.sys\\xxx";
string PASSWORD = "yyy";
Uri BASE_ADDRESS = new Uri("https://services.datafordeler.dk");
string REQUEST_URI
= "/GeoDK/topo_skaermkort/1/wms/?service=WMS&version=1.3.0
&LAYERS=dtk_skaermkort&FORMAT=IMAGE
/PNG&REQUEST=GetMap&STYLES=&CRS=EPSG:
25832&WIDTH=800&HEIGHT=545&BBOX=196364,5952066,923636,6447934
&hest100=10";

//Get SAML2 token
GenericXmlSecurityToken saml2Token;

using (var factory = new WSTrustChannelFactory(
new UserNameWSTrustBinding(SecurityMode.
TransportWithMessageCredential),
new EndpointAddress(ADFS_URL))
{
    factory.TrustVersion = TrustVersion.WSTrust13;

    factory.Credentials.UserName.UserName = USER_NAME;
    factory.Credentials.UserName.Password = PASSWORD;

    RequestSecurityToken rst = new RequestSecurityToken
    {
        RequestType = RequestTypes.Issue,
        AppliesTo = new EndpointReference(REALM),
        KeyType = KeyTypes.Bearer,
        TokenType = "urn:oasis:names:tc:SAML:2.0:assertion",
    };

    IWSTrustChannelContract channel = factory.CreateChannel();
    saml2Token = channel.Issue(rst) as GenericXmlSecurityToken;
}

//Convert token
string saml2TokenBase64 =
    Convert.ToBase64String(
        System.Text.Encoding.UTF8.GetBytes(saml2Token.TokenXml.
OuterXml.ToCharArray()));

//request a resource with the token
HttpClient client = new HttpClient();
client.DefaultRequestHeaders.Authorization
    = new AuthenticationHeaderValue("Bearer", saml2TokenBase64);
client.BaseAddress = BASE_ADDRESS;
HttpResponseMessage httpResponseMessage = await client.GetAsync
(REQUEST_URI);
```

---

## Statuskoder

Datafordeleren returnerer standard http statuskoder.

[Til toppen](#)

[Se listen over statuskoder](#)

---