

KOMBIT – BBR

DAF TRANSITION GUIDE FOR BBR CONSUMERS

Version: 1.1
Status: Final
Approver: KOMBIT
Author: Netcompany

netcompany

Document revisions

Version	Date	Author	Status	Remarks
1.0	2019-11-28	Netcompany	Final	Document finalized
1.1	2020-01-09	Netcompany / Mikkel Søres Ibsen	Final	Updated from security zone S5 to S0

References

Reference	Title	Author	Version
[Services]	DAF: Webservices (BBR)	BBR & DAF	Newest
[Events]	DAF: Hændelser (BBR)	BBR & DAF	Newest
[FileExport]	DAF: Filudtræk (BBR) – (Not in production yet, as per 2019/11/29)	BBR & DAF	Newest
[Schemas]	DAF: Skema – filudtræk og webservices (BBR)	BBR & DAF	Newest
[Bitemporality]	DAF: Bitemporalitet (BBR)	BBR & DAF	Newest
[ContactDAF]	DAF: Kontakt	DAF	Newest
[BasicData]	Good Basic Data for Everyone (PDF)	DIGST	Newest

Table of contents

1	Introduction	4
2	DAF	4
2.1	Environments.....	4
3	Datafordeler users	4
3.1	User creation.....	5
4	BBR Services on DAF	6
4.1	General	6
4.1.1	REST Service	6
4.1.2	Differences between DAF and DAFGateway	6
4.2	Bygning	9
4.2.1	Input & Output	9
4.3	Enhed.....	9
4.3.1	Input & Output	9
4.4	Grund.....	9
4.4.1	Input & Output	10
4.5	Teknisk Anlæg	10
4.5.1	Input & Output	10
4.6	Byggesag (bbrsag)	10
4.6.1	Input & Output	10
4.7	Ejendomsrelation	11
4.7.1	Input & Output	11

1 Introduction

This document is a guide for BBR consumers to transition to using BBR data via Datafordeler (DAF). The focus on this guide is on the transition from previously using BBR's DAFGateway to using DAF – though the guide can also be used by other/new BBR consumers as well.

In the following sections an introduction is given to the purpose of DAF, how to create users on DAF, how to access BBR data on DAF, and the BBR services exposed on DAF. Additional information about BBR on DAF can be found at datafordeler.dk (see References on page 2).

2 DAF

Datafordeleren (DAF) is part of the Basic Data Programme (Grunddataprogrammet) – see [BasicData] for DIGST's English pamphlet. DAF is the common public it-solution for distributing basic data (grunddata), which Styrelsen for Dataforsyning og Effektivisering (SDFE) is responsible for. Basic data is information about Denmark, it could e.g. be data on properties, addresses, roads, companies, and people. Data on DAF is supplied by registry authorities, such as CPR-kontoret, Erhvervsstyrelsen, Geodatastyrelsen, SKAT/UFST, and SDFE. These register authorities are also responsible for maintaining data and assuring the quality and integrity of the data.

2.1 Environments

DAF has several environments – with three of which being relevant for BBR consumers (**bolded** below).

- **DAF Prod**: Production environment. Gets near real-time data updates from [BBR Prod](#).
- **DAF Test06**: Most stable test environment (~preprod). Gets near real-time data updates from [BBR PP3](#).
- *DAF Test04*: Test environment with only selective registers, not including BBR.
- **DAF Test03**: Test environment for testing new fixes/changes. Gets near real-time data updates from [BBR PP1](#).
- *DAF Dev01*: Internal development environment for KMD, with no external access.

3 Datafordeler users

DAF is operating with two types of users, namely a web-user (webbruger) and a service-user (tjenestebruger). Both of these users are needed to be able to fetch data from DAF.

- **Web-user (Webbruger)**: This type of user is used to log on to the selfservice on DAF, where service-users can be created and administrated.
- **Service-user (Tjenestebruger)**: This type of user is used to access the exposed services on DAF – meaning that it is the user your applications will use to make service calls to DAF.

3.1 User creation

In this section we will go through how users are created on DAF.

First, the **web-user** should be created. To do this visit the following links, depending on which DAF environment you're attempting to access (note: BBR is not on DAF Test04):

- DAF Prod: <https://selfservice.datafordeler.dk>
- DAF Test06: <https://test06-selfservice.datafordeler.dk>
- DAF Test03: <https://test03-selfservice.datafordeler.dk>

Then follow the following steps:

1. Click the 'OPRET'-button.
2. Choose whether the user should be created by username and password (Brugernavn / Adgangskode) or by Certificate (Certifikat).
3. If username and password is chosen:
 - a. A dialog is shown, where a consent should be given by clicking the 'JA'-button (yes). And the same goes for the terms of use, which will be shown in the subsequent dialog.
 - b. Now, information like name, address, contact, and password should be filled.
 - c. Click 'OPRET BRUGER' to finish the creation, and then the page will be redirected to the log on page.
4. If Certificate is chosen:
 - a. A dialog is shown, where a consent should be given by clicking the 'JA'-button (yes). And the same goes for the terms of use, which will be shown in the subsequent dialog.
 - b. Choose NemLog-in. The page will be redirected to NemLog-in. Click on the tab 'Log på med nøglefil' and log in.
 - c. The page will now be redirected back to selfservice (selvbetjeningen).
 - d. Now, information like name, address, contact, and password should be filled.
 - e. Click 'OPRET BRUGER' to finish the creation, and then the page will be redirected to the log on page.

Secondly, the **service-user** needs to be created. To do this login with your web-user on <https://selfservice.datafordeler.dk/> (or relevant test environment).

1. Go to the section 'BRUGERE' and click the plus-button next to 'WEBBRUGER'.



2. Here it can be chosen whether the user should be created by Username / Password (Brugernavn / Adgangskode) or Certificate (Certifikat). **Note**, to access BBR data on DAF certificates should be used, hence **the service-user accessing**

BBR data should be created choosing certificate.

3. If Certificate is chosen – either FOCES (function certificate) or VOCES (company certificate).
 - a. The new service-user is created and shown next to the tab 'WEBBRUGER'.
 - b. Find your MOCES (Employee) certificate and upload the FOCES or VOCES certificate.

- c. The service-user will be updated with the certificate.
4. If Username / Password is chosen.
 - a. The system will generate a username, which will be shown next to the tab 'WEBBRUGER'.
 - b. Fill the password fields and save.

4 BBR Services on DAF

This section is describing the BBR services that can be called on DAF.

4.1 General

This section describes access level, security zone, URL structure for the REST services, and differences between DAF and the DAFGateway on BBR, which will be phased out.

4.1.1 REST Service

- Format: XML/JSON
- Security zone: 0
- Access level: Users will have to apply for access (see [Applying for access to BBR](#))
- Structure: <endpoint>/REGISTRY/SERVICE/VERSION/SERVICETYPE/METHOD?format=FORMAT
 - o Example: <https://services.datafordeler.dk/BBR/BBRPublic/1/REST/bygning?format=json>

Although the security zone is S0, the service is only available to known users and should be called with either a certificate or with a username and password.

4.1.2 Differences between DAF and DAFGateway

In this section some of the more notable differences between DAF and BBR's own DAFGateway are listed:

- **Change of URL:** Users should use the URL to DAF <https://services.datafordeler.dk/BBR/BBRPublic/1/REST/> instead of the URL to DAFGateway <https://sg.bbr.dk/External/DAFGatewayV3/> – and for parameters, DAFGateway always takes “apikey” as input and DAF takes “format” (json or xml) as input.
 - **DAFGateway example:**
<https://sg.bbr.dk/External/DAFGatewayV3/bygning?apikey=APIKEY&pagesize=1>
 - **DAF example:**
<https://services.datafordeler.dk/BBR/BBRPublic/1/REST/bygning?format=json&pagesize=1>
- **Multiple values for parameter:** When searching for multiple values (e.g. Status), the character separating values are now ‘|’ (e.g. 6|9), where on the DAFGateway it was ‘,’ (e.g. 6,9).
 - **DAFGateway example:**
<https://sg.bbr.dk/External/DAFGatewayV3/bygning?apikey=APIKEY&status=6,9>
 - **DAF example:**
<https://services.datafordeler.dk/BBR/BBRPublic/1/REST/bygning?format=json&status=6|9>
- **datafordelerOpdateringstid:** DAF returns an extra field, namely datafordelerOpdateringstid, which contains information on when the data has been updated on DAF. “datafordelerOpdateringstid” is always the first attribute.
 - **DAFGateway example:**
`{"byg007Bygningsnummer": 11, "byg021BygningensAnvendelse": "420", ...`
 - **DAF example:**
`{"datafordelerOpdateringstid": "2019-05-12T03:44:46.830256+02:00", "byg007Bygningsnummer": 11, "byg021BygningensAnvendelse": "420", ...`

- **Null and 0 values:** Whenever a field has not been filled (null) the DAFGateway filled it with a null value. Whenever an Integer field has the value 0, the DAFGateway returned the 0 value. DAF does not return fields that have not been filled (null) or integers filled with 0.
 - **DAFGateway example:**

```
{ "byg007Bygningsnummer": null, "byg021BygningensAnvendelse": "510",
  "byg024AntalLejlighederMedKøkken": 0,
  "byg025AntalLejlighederUdenKøkken": 0, "byg026Opførelsesår": null, ... }
```
 - **DAF example:**

```
{ "datafordelerOpdateringstid": "2019-05-12T13:44:19.240767+02:00",
  "byg021BygningensAnvendelse": "510", ... }
```
- **Jordstykker:** Jordstykker should be fetched from MU/MAT by using an ID of the type Integer, and no longer from BBR with an ID with the type UUID.
 - **DAFGateway example for Grund->Jordstykke:**

```
{ ... jordstykkeList": [ "897cfb05-0aa7-418d-8dc7-11075c45f252" ] ... }
```

->

<https://sg.bbr.dk/External/DAFGatewayV3/jordstykke?apikey=APIKEY&id=897cfb05-0aa7-418d-8dc7-11075c45f252>
 - **DAF example:**

```
{ ... jordstykkeList": [ "1928359" ] ... }
```

-> (REST example)

<https://certservices.datafordeler.dk/Matrikel/Matrikel/1/rest/SamletFastEjendom?format=json&JordstykkeId=1928359>
 - REST JSON schema ("samletfastejendom"):

<https://confluence.datafordeler.dk/pages/viewpage.action?pageId=13664474>
 - WFS documentation:

<https://datafordeler.dk/dataoversigt/matriklen-mat/matriklen-med-historik/>
- **Ejendomsrelation:** The service on DAFGateway has the field 'ejendomsType' with type Integer, where DAF has the field 'ejendomstype' with type string. The value is the same, it is only the type that is different.
 - **DAFGateway example:**

```
{ ... "ejendomsType": 1 ... }
```
 - **DAF example:**

```
{ ... "ejendomstype": "1" ... }
```
- **Nested lists:** Services on DAFGateway with nested lists (e.g. etageList in Bygning), shows an ID-attribute called 'id' and the related object. On DAF, the ID-attribute is called 'id_lokalId'.
 - **DAFGateway example:**

```
"etageList": [ {
  "id": "a2857672-e967-47de-a36f-d06dc9180fc2",
  "etage": { ... }
} ]
```
 - **DAF example:**

```
"etageList": [ {
  "id_lokalId": "a2857672-e967-47de-a36f-d06dc9180fc2",
  "etage": { ... }
} ]
```


4.2 Bygning

On DAF you can fetch information on a BBR Bygning by using the URL: [.<endpoint>/BBR/BBRPublic/1/REST/bygning](https://services.datafordeler.dk/BBR/BBRPublic/1/REST/bygning)

4.2.1 Input & Output

The following parameters can be appended to the URL to fetch a specific Bygning.

- Id (List<UUID>), VirkningFra (DateTime), VirkningTil (Datetime), Virkningsaktoer (String), RegistreringFra (DateTime), RegistreringTil (DateTime), Registreringsaktoer (String), Status (List<String>), Forretningsproces (String), Forretningsomraade (String), Forretningshaendelse (String), KommuneKode (String), MedDybde (Boolean), DAFTimestampFra (DateTime), DAFTimestampTil (DateTime), BFENummer (Integer), Etage (List<UUID>), Fordelingsareal (UUID), Opgang (List<UUID>), TekniskAlaeg (List<UUID>), Ejendomsrelation (UUID), AdresseIdentifier (UUID), Bygning (List<UUID>), PeriodeaendringFra (DateTime), PeriodeaendringTil (DateTime), KunNyestePeriode (Boolean).

An example on a query could be:

<https://services.datafordeler.dk/BBR/BBRPublic/1/rest/bygning?&format=json&pagesize=1&id=06558f63-f742-440a-97e5-90f880250bf0>

The output schema can be found here:

<https://confluence.datafordeler.dk/pages/viewpage.action?pageId=13664880> (see "Bygning.schema.json")

4.3 Enhed

On DAF you can fetch informations on a BBR Enhed by using the URL: [.<endpoint>/BBR/BBRPublic/1/REST/enhed](https://services.datafordeler.dk/BBR/BBRPublic/1/REST/enhed)

4.3.1 Input & Output

The following parameters can be appended to the URL to fetch a specific Enhed.

- Id (List<UUID>), VirkningFra (DateTime), VirkningTil (Datetime), Virkningsaktoer (String), RegistreringFra (DateTime), RegistreringTil (DateTime), Registreringsaktoer (String), Status (List<String>), Forretningsproces (String), Forretningsomraade (String), Forretningshaendelse (String), KommuneKode (String), MedDybde (Boolean), DAFTimestampFra (DateTime), DAFTimestampTil (DateTime), BFENummer (Integer), Etage (List<UUID>), Fordelingsareal (UUID), Opgang (List<UUID>), TekniskAlaeg (List<UUID>), Ejendomsrelation (UUID), AdresseIdentifier (UUID), Bygning (List<UUID>), PeriodeaendringFra (DateTime), PeriodeaendringTil (DateTime), KunNyestePeriode (Boolean).

An example on a query could be:

<https://services.datafordeler.dk/BBR/BBRPublic/1/rest/enhed?&format=json&pagesize=1&id=27673802-d218-43b6-a0d9-8e2dfe589d02>

The output schema can be found here:

<https://confluence.datafordeler.dk/pages/viewpage.action?pageId=13664880> (see "Enhed.schema.json")

4.4 Grund

On DAF you can fetch informations on a BBR Grund by using the URL: [.<endpoint>/BBR/BBRPublic/1/REST/grund](https://services.datafordeler.dk/BBR/BBRPublic/1/REST/grund)

4.4.1 Input & Output

The following parameters can be appended to the URL to fetch a specific Grund.

- Id (List<UUID>), VirkningFra (DateTime), VirkningTil (Datetime), Virkningsaktoer (String), RegistreringFra (DateTime), RegistreringTil (DateTime), Registreringsaktoer (String), Status (List<String>), Forretningsproces (String), Forretningsomraade (String), Forretningshaendelse (String), KommuneKode (String), MedDybde (Boolean), DAFTimestampFra (DateTime), DAFTimestampTil (DateTime), BFENummer (Integer), Bygning (UUID), TekniskAnlaeg (UUID), Ejendomsrelation (UUID), Jordstykke (Integer), Husnummer(List<UUID>), PeriodeaendringFra (DateTime), PeriodeaendringTil (DateTime), KunNyestePeriode (Boolean).

An example on a query could be:

<https://services.datafordeler.dk/BBR/BBRPublic/1/rest/grund?&format=json&pagesize=1&id=e396f856-52d4-4479-8ec1-6433e472ccfd>

The output schema can be found here:

<https://confluence.datafordeler.dk/pages/viewpage.action?pageId=13664880> (see "Grund.schema.json")

4.5 Teknisk Anlæg

On DAF you can fetch informations on a BBR Enhed by using the URL: [endpoint>/BBR/BBRPublic/1/REST/tekniskanlaeg](https://services.datafordeler.dk/BBR/BBRPublic/1/REST/tekniskanlaeg)

4.5.1 Input & Output

The following parameters can be appended to the URL to fetch a specific Teknisk Anlæg.

- Id (List<UUID>), VirkningFra (DateTime), VirkningTil (Datetime), Virkningsaktoer (String), RegistreringFra (DateTime), RegistreringTil (DateTime), Registreringsaktoer (String), Status (List<String>), Forretningsproces (String), Forretningsomraade (String), Forretningshaendelse (String), KommuneKode (String), MedDybde (Boolean), DAFTimestampFra (DateTime), DAFTimestampTil (DateTime), Bygning (UUID), Enhed (UUID), Grund (UUID), BFENummer (Integer), Nord (Double), Syd (Double), Oest (Double), Vest (Double), PeriodeaendringFra (DateTime), PeriodeaendringTil (DateTime), KunNyestePeriode (Boolean).

An example on a query could be:

<https://services.datafordeler.dk/BBR/BBRPublic/1/rest/tekniskanlaeg?&format=json&pagesize=1&id=6f074d7d-4389-4f69-8963-4656fcacfb8c>

The output schema can be found here:

<https://confluence.datafordeler.dk/pages/viewpage.action?pageId=13664880> (see "TekniskAnlæg.schema.json")

4.6 Byggesag (bbrsag)

On DAF you can fetch informations on a BBR Byggesag by using the URL: [endpoint>/BBR/BBRPublic/1/REST/bbrsag](https://services.datafordeler.dk/BBR/BBRPublic/1/REST/bbrsag)

4.6.1 Input & Output

The following parameters can be appended to the URL to fetch a specific Byggesag.

- Id (List<UUID>), VirkningFra (DateTime), VirkningTil (Datetime), Virkningsaktoer (String), RegistreringFra (DateTime), RegistreringTil (DateTime), Registreringsaktoer (String), Status (List<String>), Forretningsproces (String), Forretningsomraade (String), Forretningshaendelse (String), KommuneKode (String), MedDybde (Boolean),

DAFTimestampFra (DateTime), DAFTimestampTil (DateTime), Bygning (UUID), Enhed (UUID), Etage (UUID), Grund (UUID), Opgang (UUID), TekniskAnlaeg (UUID), PeriodeaendringFra (DateTime), PeriodeaendringTil (DateTime), KunNyestelPeriode (Boolean).

An example on a query could be:

<https://services.datafordeler.dk/BBR/BBRPublic/1/rest/bbrsag?&format=json&pagesize=1&id=df57af50-ab44-4ead-87ce-9ea304cc596d>

The output schema can be found here:

<https://confluence.datafordeler.dk/pages/viewpage.action?pageId=13664880> (see "BBRSag.schema.json")

4.7 Ejendomsrelation

On DAF you can fetch informations on a BBR Byggesag by using the URL: <endpoint>/BBR/BBRPublic/1/REST/ejendomsrelation

4.7.1 Input & Output

The following parameters can be appended to the URL to fetch a specific Ejendomsrelation.

- Id (List<UUID>), VirkningFra (DateTime), VirkningTil (Datetime), Virkningsaktoer (String), RegistreringFra (DateTime), RegistreringTil (DateTime), Registreringsaktoer (String), Status (List<String>), Forretningsproces (String), Forretningsomraade (String), Forretningshaendelse (String), KommuneKode (String), MedDybde (Boolean), DAFTimestampFra (DateTime), DAFTimestampTil (DateTime), BFENummer (Integer), BPF (Integer), Ejerforholdskode (String), Ejerlejlighed (Integer), Ejendomsnummer (Integer), SamletFastEjendom (Integer), Vurderingsejendomsnummer (Integer), PeriodeaendringFra (DateTime), PeriodeaendringTil (DateTime), KunNyestelPeriode (Boolean).

An example on a query could be:

<https://services.datafordeler.dk/BBR/BBRPublic/1/rest/ejendomsrelation?&format=json&pagesize=1&bfnummer=6004660>

The output schema can be found here:

<https://confluence.datafordeler.dk/pages/viewpage.action?pageId=13664880> (see "Ejendomsrelation.schema.json")