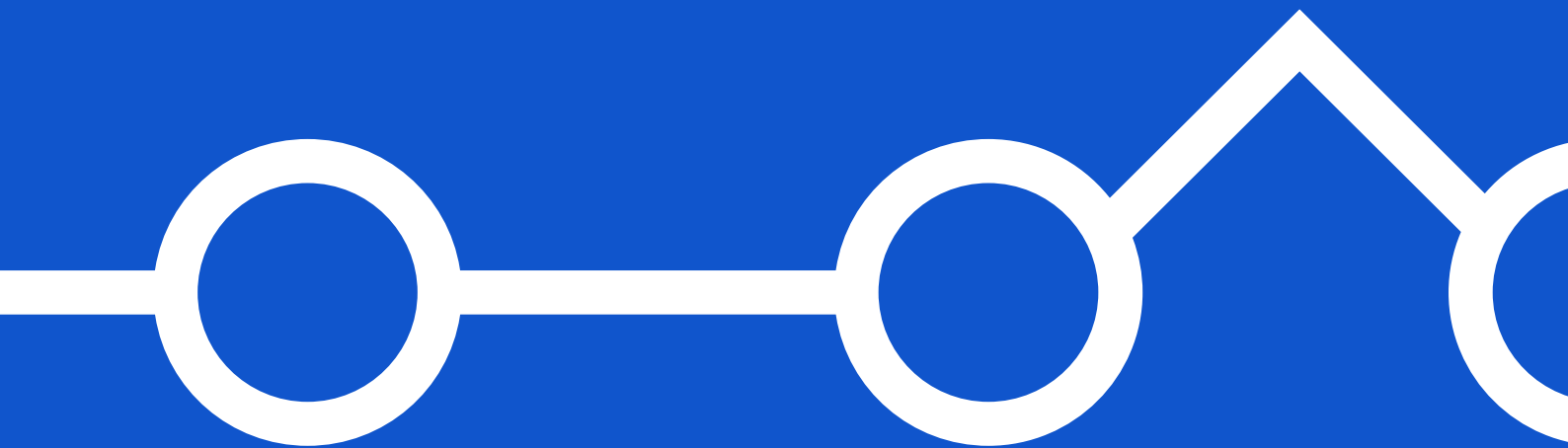


DIGITALISERINGS KATALOGET

SF1590_B_01 AFSEND DEBITORREGISTRERING

PROGRAMMER'S GUIDE

DECEMBER 2025



KOMBIT

Kommunernes it-fællesskab

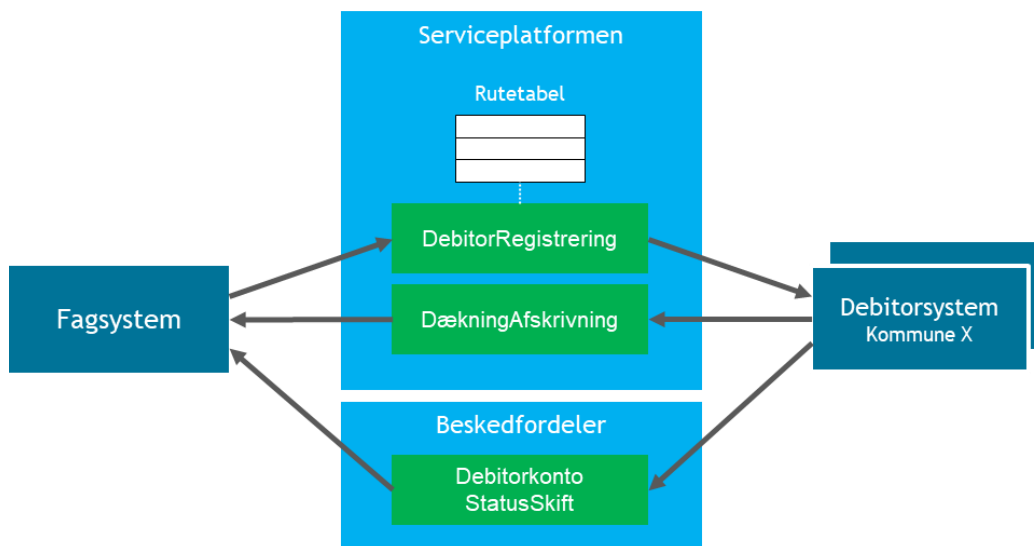
Indholdsfortegnelse

1. INTRODUKTION	4
2. KLASSIFICEREDE ELEMENTER	5
2.1. Introduktion	5
2.2. HavneAdm Debtorsortiment.....	5
3. WEBSERVICES GENERELT	9
3.1. Hovedoplysninger	9
3.2. Note vedrørende Revision.....	Fejl! Bogmærke er ikke defineret.
4. SERVICE DEBITORREGISTRERING	FEJL! BOGMÆRKE ER IKKE DEFINERET.
4.1. Dokumentation	10
4.2. Konfiguration	10
4.3. Test mod debtorsystem	12
4.4. WSDL/XSD encoding	12
4.5. Operation OpretDebitorkontoRegistrering	12
4.5.1. Introduktion	12
4.5.2. DebitorkontoFelter.....	13
4.5.3. FordringListe	14
4.5.4. Øvrige elementer	16
4.5.5. Eksempel	16
4.6. Operation SætFordringTilOpkrævning	19
4.7. Operation AfslutHæftelse.....	19
4.8. Operation ForetagEjerskifte.....	20
4.9. Operation OverdragDebitorkonto	Fejl! Bogmærke er ikke defineret.
4.10. Operation AfslutAdministrator	21
5. SERVICE DEBITORKONTOSTATUSSKIFT	22
5.1. Dokumentation	22
5.2. Konfiguration	22
5.3. Eksempelbesked	24
6. SERVICE DÆKNINGAFSKRIVNING	29
6.1. Dokumentation	30

6.2. Implementer sikkerhed	30
6.3. Registrer service i FK ADM	31
6.4. Bestil gennemstilling på Serviceplatformen	36
6.5. Bestilling af webservice rute	Fejl! Bogmærke er ikke defineret.
6.6. Operation Ping	36
6.7. Operation OpretDækningAfskrivning	39
6.7.1. Validering	39
6.7.2. Generel leverancekontrol	39
6.7.3. Forretningsmæssig validering	41
6.7.4. Eksempel - dækning	41
6.7.5. Test validering	43
VERSIONSHISTORIK	44

1. INTRODUKTION

Denne vejledning henvender sig til leverandører, der skal implementere [SF1590 B 01 Afsend DebitorRegistrering](#) som fagsystem. Integrationen består af tre flows, markeret med grønt på følgende figur:



For anvendelse af de fælleskommunale komponenter henvises til de pågældende [Kom-godt-i-gang vejledninger](#) (KGIG).

1. DebitorRegistrering

Fagsystemet sender fordringer til økonomisystemer ved at kalde webservice *DebitorRegistreringAfsend* på Serviceplatformen. Du skal etablere en webservice-klient.

2. DækningAfskrivning

Fagsystemet modtager dækningsafskrivninger fra økonomisystemer via Serviceplatformen. Du skal etablere og udstille en webservice.

3. DebitorkontoStatusSkift

Fagsystemet modtager hændelser ved statusskifte på konti. Du skal integrere til Beskedfordeler.

Venligst se de respektive [Kom-godt-i-gang](#) vejledninger mht. tekniske forudsætninger; bestilling af funktionscertifikat, registrering af anvendelsesystem, adgang til administrative brugerflader, anmodning om serviceaftaler samt bestilling af ruter. Bemærk, at de tre integrationsflows benytter forskellige teknologier og komponenter, så de har hver egne forudsætninger.

Hent dokumentationspakken for [SF1590_B_01 Afsend DebitorRegistrering](#). I denne finder du den grundlæggende integrationsbeskrivelse samt en zip-fil for hvert integrationsflow indeholdende en Servicebeskrivelse, Servicekontrakter og eventuelle bilag.

2. KLASSIFICEREDE ELEMENTER

2.1. Introduktion

Grundlæggende elementer i datamodellen er såkaldte *klassificerede elementer*, og det er vigtigt, at du læser afsnittet i *DebitorRegistreringAfsend* Servicebeskrivelsen vedrørende disse, og har forstået hvordan det fungerer. Hvert fagsystem skal oprette eget debtorsortiment og definere egne værdier for de klassificerede elementer. Det er en forudsætning, at du har oprettet dette, samt at debtorsystemet, du integrerer til, har registreret disse jf. integrationsbeskrivelsen: *Debtorsystemet skal forud for idriftsættelse af fagsystemets integration etablere abonnement på fagsystemets sortiment, så dette er kendt og mapet i omsætningstabeller* - det sker pt. rent praktisk, ved at kommunen bestiller abonnementet i FK KLA brugerfladen.

2.2. HavneAdm Debtorsortiment

Til denne vejledning benytter vi sortiment *HavneAdm Debtorsortiment* som eksempel. Du skal selvfølgelig benytte værdier fra dit eget fagsystems sortiment. Vi har aflæst værdierne fra FK Klassifikation brugerfladen. Her eksempel på Debitorkontotype *Havn*:

▼ Havn	Klasserreference	
— Havneafgifter	UUID	591f8439-6bae-49a2-9928-7b0f97831e90
— Havneydelser	Ejer (CVR)	KOMBIT A/S (19435075)
	Brugervendt nøgle	Havn
	Titel	Havn
	Beskrivelse	Debitorkontotype til fordringer på havneområdet

Til hver *DebitorkontoType* angives en eller flere *FordringkravType*, her eksempel med *Havneafgifter*:

<ul style="list-style-type: none"> ▼ Debitorkontotyper ▼ Havn <li style="background-color: #2e8b57; color: white; padding: 2px;">- Havneafgifter - Havneydelser 	<table border="1"> <thead> <tr style="background-color: #2e8b57; color: white;"> <th colspan="2">Klasserference</th> </tr> </thead> <tbody> <tr> <td>UUID</td> <td>496dee04-de3a-4d77-bf17-c47de23b2a47</td> </tr> <tr> <td>Ejer (CVR)</td> <td>KOMBIT A/S (19435075)</td> </tr> <tr> <td>Brugervendt nøgle</td> <td>Havneafgifter</td> </tr> <tr> <td>Titel</td> <td>Havneafgifter</td> </tr> <tr> <td>Beskrivelse</td> <td>Fordringstype for offentlige havneafgifter, kanal, vandvejs- og lodsafgifter</td> </tr> </tbody> </table>	Klasserference		UUID	496dee04-de3a-4d77-bf17-c47de23b2a47	Ejer (CVR)	KOMBIT A/S (19435075)	Brugervendt nøgle	Havneafgifter	Titel	Havneafgifter	Beskrivelse	Fordringstype for offentlige havneafgifter, kanal, vandvejs- og lodsafgifter
Klasserference													
UUID	496dee04-de3a-4d77-bf17-c47de23b2a47												
Ejer (CVR)	KOMBIT A/S (19435075)												
Brugervendt nøgle	Havneafgifter												
Titel	Havneafgifter												
Beskrivelse	Fordringstype for offentlige havneafgifter, kanal, vandvejs- og lodsafgifter												

Se i øvrigt video [her](#), om hvordan du opretter dit fagssystem-sortiment.

Det er værdien i *Klasserference* vi skal benytte. Bemærk, at *HavneAdm Debtorsortiment* blot indeholder eksempler på de mest anvendte klassificerede elementer, det er ikke komplet. Vi etablerer en liste af værdier for klassificerede elementer, vi skal benytte i vores kode:

- BilagFilType (n/a)
- BilagMaterialeType (n/a)
- DebitorkontoidentifikationType
 - da335bc8-28bd-408f-b7b2-a10a03a0f8f3 CPR-nummer
 - ad9cc9c9-1c6f-47d9-8987-40bf207ebdc7 CVR-nummer
 - 9fa7d141-7a8e-4b32-bd3c-86e025d30747 ErstatningsCPR-nummer
- DebitorkontoType
 - 591f8439-6bae-49a2-9928-7b0f97831e90 Havn
 - 496dee04-de3a-4d77-bf17-c47de23b2a47 Havneafgifter (FordringKravType)
 - 33fec3d7-fae0-4ab1-89e2-81f195f08fc2 Havneydelser (FordringKravType)
- DebitorkontoFølsomhed (se kommentar forned)
- 292e85a9-8ad4-46df-9e50-f97d6837ad74 Almindelige personoplysninger
- DebitorkontoOrganisatoriskReference
 - 4053d58d-bd5e-48d9-b9d8-2c041faeccd5 Team 1 (3)
- DebitorkontoSpecifikationType (n/a) (se kommentar forned)
- DebitorkontoSpecifikationKlasseVærdi (n/a)
- FordringEnhedsoplysningEnhed (n/a)
- FordringFølsomhed (se kommentar forned)
- FordringNedreguleringÆndringÅrsagKode
 - 78503485-8fa5-476a-8881-b76693e74c81 Anden årsag
 - 60b1e324-4242-4ee7-b304-13245e104c7e Nedregulering af fordringen
- FordringSpecifikationType (se kommentar forned)
 - 719ba471-5394-42ad-898d-89c920b1d0c2 Inddrivelsestekst (1)

- b0598db0-5b2a-4112-bc84-f20219e96861 PSRM Kravtype (1,2) angives i elementet *FordringSpecifikationKlasseVærdi*
 - 1fb4083a-b469-41ef-a29e-928a42b11552 Offentlige havneafgifter, ...
 - 83032492-33c6-4521-880e-7dbb2cfc15a8 Havneydelser
- MaterialeabonnementType (n/a)
- ØIRPartIdType
 - 41a31ac1-27c0-4ef1-be80-175f7dd0de65 CPR-nummer
 - 7f1b3294-d216-4bd5-be36-e1812647ef6e ErstatningsCPR-nummer
 - 425a09c9-7ab6-4fe9-83d2-88ab454bddaf CVR-nummer

(1) Benyttes når fordringen skal kunne inddrives af Gældsstyrelsen

(2) PSRM = Fordringstyper defineret i Gældsstyrelsens inddrivelsessystem

(3) Organisatoriske referencer oprettes i jeres sortiment af kommunen!

N/a angiver, at der ikke er defineret eksempler på pågældende klassificerede element i *HavneAdm Debtorsortiment*.

Vedrørende *DebitorkontoFølsomhed* og *FordringFølsomhed*; her skal du benytte de predefinerede værdier fra Klassifikation:

1d81c472-0808-44cc-963d-f5ef0170ae1d	Ikke fortrolige data
292e85a9-8ad4-46df-9e50-f97d6837ad74	Fortrolige personoplysninger eller fortrolige forretningsdata
31c09910-e011-46a5-86fb-254374421fe8	Følsomme personoplysninger eller følsomme forretningsdata
44f4108b-26d4-46de-a90f-35e35b55b8d8	Særligt beskyttede oplysninger

Du opretter de følsomhed-klasser, du har brug for, i dit eget sortiment. Du behøver ikke oprette alle fire. Bemærk, at værdierne starter med 1, 2, 3 og 4 og at man dermed nemt ud fra det første tal kan se angivet følsomhedsniveau.

Vedrørende *DebitorkontoSpecifikationType*, se [IB] afsnit *Integrationens predefinerede Debitorkontospecifikationstyper*:

Vedrørende *FordringSpecifikationType*, se [IB] afsnit *Integrationens predefinerede Fordringsspecifikationstyper*. De to predefinerede benyttes, når fordringen skal kunne inddrives af Gældsstyrelsen:

b0598db0-5b2a-4112-bc84-f20219e96861 PSRM_Kravtype
719ba471-5394-42ad-898d-89c920b1d0c2 Inddrivelsestekst

Bemærk, at *Klassereference* i FK KLA brugerfladen modsvarer:

- "klasseUuid" i JSON-udtræk af sortiment
- "Klassidentifikation" i XML-udtræk man modtager på Sortiment abonnement.

På DebitorkontoType *Havn* ser vi angivet en Regel:

Information om regel findes i Servicebeskrivelsen:

Havn

✎ Rediger
✕ Slet
📄 Vis historik

Registreringsværdi	
UUID	9a428f49-5e33-4a80-93fb-c83e5076ee9c
Redaktør (CVR)	KOMBIT A/S (19435075)
Regel	Alle fordringer på debitorkontoen skal have PSRM-fordringstype (0e13d7e8-79aa-4a2d-9ba7-f8971c227ecc)

9.1 Anvendte Regler på Debitorkontotype

Regelidentifikation	Indeholder detaljering, med mulige Krav-Type(r) på Fordring. Klassidentifikation benyttes ved registrering af KravType	Alle fordringer under debitorkontototen skal have samme OpkrævningTilstand	Krav om angivelse af FordringSpecifikation-Typerne PSRM-krav-type og Inddrivelses-tekst	Krav om angivelse af DebitorkontoSpecifikationTyperne Udbetalingssparagraf og Tilbagebetalingsparagraf
26b8ff2e-528b-462b-bf01-74d441c0ad18	✓			
227f689a-cc8c-4490-9c37-8b51940cb495	✓			✓
0e13d7e8-79aa-4a2d-9ba7-f8971c227ecc	✓		✓	
356b7807-cc2f-479e-b39c-021d73f8700e	✓		✓	✓
ea3a95a2-a797-47c3-b245-825fc94b1e45	✓	✓		
e80e422f-ec15-4111-b945-762cac3960a3	✓	✓		✓
890df1fd-b9b3-4656-893f-12e13af95674	✓	✓	✓	
e6f51517-94ab-466c-b3e3-7242aaaa0ebd	✓	✓	✓	✓

Tabel 10 - Regler på Debitorkontotype

Den første regel "Indeholder detaljering, med mulige *KravType(r)* på *Fordring*. *Klassidentifikation* benyttes ved registrering af *KravType*" betyder, at *DebitorkontoType Havn* har tilknyttet én til flere *FordringkravType* - hvilket fremgår af *HavneAdm* sortiment.

Den anden regel "Krav om angivelse af *FordringSpecifikationTyperne PSRM-kravtype* og *Inddrivelsestekst*" betyder, at når *DebitorkontoType Havn* angives, da skal alle fordringer angives med *FordringSpecifikation* -> *FordringSpecifikationType PSRM* samt *Inddrivelsestekst*. Dette er illustreret i det praktiske eksempel, der følger senere i vejledningen.

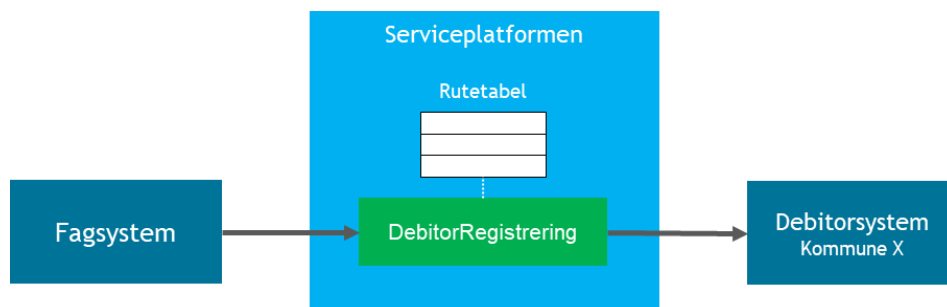
3. WEBSERVICES GENERELT

3.1. Hovedoplysninger

I Digitaliseringskataloget [Teknik - Webservice](#) finder du praktisk information om sikkerhedsmodellen, heriblandt detaljer om Hovedoplysninger i dokumentet *Transaktionsspor og fejlbehandling ved servicekald*. Den fulde struktur er:

```
<HovedOplysninger>
  <TransaktionsId>?</TransaktionsId>
  <TransaktionsTid>?</TransaktionsTid>
  <AuthorityContext>
    <MunicipalityCVR>?</MunicipalityCVR>
  </AuthorityContext>
  <Rute>
    <AfsenderOrganisation>?</AfsenderOrganisation>
    <AfsenderItSystemInstans>?</AfsenderItSystemInstans>
    <ModtagerOrganisation>?</ModtagerOrganisation>
    <!--Optional:-->
    <ModtagerItSystemInstans>?</ModtagerItSystemInstans>
  </Rute>
</HovedOplysninger>
```

- *AuthorityContext* skal ikke angives, hvis der benyttes token-sikkerhed.
- Værdier for *AfsenderItSystemInstans* samt *ModtagerItSystemInstans* aflæses i ADM (UUID).
- *ModtagerItSystemInstans* skal ikke angives af fagsystemer.



Fagsystemet sender en Debitorregistrering til debitorsystemet via Serviceplatformen.

3.2. Dokumentation

De essentielle dele af dokumentationspakken er Servicebeskrivelsen, Servicekontrakten og Informationsmodellen:

Web service DebitorRegistreringAfsend_3.0.zip

[SB] Servicebeskrivelse til DebitorRegistreringAfsend_3.0.pdf

WSDL og XSDs til DebitorRegistreringAfsend_3.0.zip

[SK] \package\DebitorRegistreringAfsendServiceToken.wsdl

Bilag til SF1590_B_01 Integrationsbeskrivelse til Afsend DebitorRegistrering_4.0.zip

[IM] Informationsmodel for dataudveksling med Debitor-domænet_1.7.pdf

3.3. Konfiguration

Registrering af IT-system

Vores IT-system er registreret i ADM og her ser du UUID, som du får brug for senere. Her eksempel med vores testsystem:

Fælleskommunalt Administrationsmodul

Opgaveoversigt
Organisationer
It-systemer
Serviceaftaler
Føderationsaftaler
Jobfunktionsroller

KDI CTT Test System #2

Stamdata Dataafgrænsningstyper Anvendersystem

UUID: f269ca0c-7b5d-49e2-a585-1ee39ba5a431

Vi har aktiveret typen "Anvendersystem" under stamdata:

Type:

- Anvendersystem**
- Brugervendt system
- Identity Provider
- Serviceudbyder

Vær opmærksom på at du ikke kan vælge systemtyper fra, når først du har gemt.



Og under faneblad "Anvendersystem" uploadet den offentlige version af vores funktionscertifikat:


KDI CTT Test System #2

Stamdata Dataafgrænsningstyper **Anvendersystem** Brugerve...

Betroet: Må impersonere

Certifikat: *

 Træk fil med PEM indhold fra systemcertifikat herind 

Certifikat	Udløb ^	
KDI STS SFTP IBA Test2 (funktio...	2023-02-20	

Servicedetaljer

Du finder dem, ved at klikke på "+" udfor denne i [Digitaliseringskataloget](#):

Webservice / Serviceplatformen Version **I DRIFT** +

DebitorRegistreringAfsend 3.0

Eksternt testmiljø

Endpoint: <https://exttest.serviceplatformen.dk/service/OIR/DebitorRegistreringAfsend/3>

UUID: d9f6680a-a636-486b-b8ae-1ea8dc65e26c

EntityId: <http://oir.serviceplatformen.dk/service/debitorregistreringafsend/3>

Hvis du benytter token-sikkerhed i kaldet til Serviceplatformen (Liberty Basic SOAP Binding), da signeres svar med certifikat "sp-signing". Security Token Service signerer svar med "adgangsstyring" certifikatet. Du finder certifikater benyttet i den fælleskommunale infrastruktur [her](#).

Service navn i ADM, når du skal anmode om serviceaftale: "DebitorRegistreringAfsend V3.0".

Du skal også anmode om serviceaftale til debtorsystemets service.

3.4. Test mod debtorsystem

Du skal aftale testforløb med kommunen og ansvarlige for debtorsystemet.

Yderligere har KDI etableret et virtuelt debtorsystem man kan benytte til de initiale tests. Venligst se [Test - SF1590_B_01](#):

Til testformål tilbydes en virtualisering af et ERP-systems udstilling af SF1590_B_01 DebitorRegistreringAfsend. Virtualiseringen er i stand til at returnere forskellige fejlkoder, som et resultat af en angivet kode i processing af elementet. Du kan læse mere i vejledningen 'SF1590_B_01 DebitorRegistreringAfsend – Vejledning til test med Virtualiseringen'. Der findes også en pakke med test eksempler. Pakken indeholder udvalgte eksempler på kald og svar fra servicevirtualiseringerne.

3.5. WSDL/XSD encoding

Webservicekontrakter er UTF-8 kodet uden Byte-Order-mark (BOM). På Windows-platformen skal du tilføje BOM til filerne, da de ellers ikke identificeres som UTF-8 filer. Dette gøres med en Powershell kommando som følger. Vi har kopieret kontrakten til en temp-folder og kører først script på xsd-filerne:

```
PS C:\temp\SF1590Bwsdl> Get-ChildItem *.* -Recurse | ForEach-Object {  
    $content = Get-Content -Path $_  
    Set-Content -Path $_.Fullname -Value $content -Encoding UTF8 -PassThru -Force  
}
```

Nu er der tilføjet BOM til filerne og vi kan tilføje reference til service fra Visual Studio eller Java.

3.6. Operation OpretDebitorkontoRegistrering

3.6.1. Introduktion

Inputdata <OpretDebitorkontoRegistrering_I> til operationen har følgende elementer:

HovedOplysninger	Skal altid udfyldes - se Hovedoplysninger .
------------------	---

EjerskifteMarkering	Angives kun, hvis operationen foretages i forbindelse med ejerskifte af en debitorkonto. Sekvensen er beskrevet i afsnit "ForetagEjerskifte [Operation]" i servicebeskrivelsen. Se også uddybning i afsnit <u>Operation ForetagEjerskifte</u> .
DebitorkontoUnikIdentifikation	Skal altid udfyldes. Angives med ny unik værdi ved oprettelse af konto og samme værdi efterfølgende ved opdateringer til konto.
DebitorkontoFelter	Udfyldes kun ved oprettelse af konto (et nyt DebitorkontoUnikIdentifikation ikke allerede kendt af debtorsystemet)

JuridiskAnsvarligPartAdresse	Fælles for alle elementer til venstre er, at de kun udfyldes hvis de skal oprettes Hvis <DebitorkontoFelter> er angivet (oprettelse af en Debitorkonto), da skal der der altid også angives mindst én <FordringListe><Fordring>. Hvis <DebitorkontoFelter> <u>ikke</u> er angivet (opdatering af en Debitorkonto) - og ingen af de andre elementer er angivet, da skal der som minimum være angivet en <FordringListe><Fordring> eller <FordringNedreguleringListe><FordringNedregulering>.
Administrator	
JuridiskMedhæfterListe	
FordringListe	
HæftelseListe	
FordringNedreguleringListe	
OpkrævningAnmodningListe	
OpkrævningListe	

Dette afsnit beskriver, hvordan du opretter en debitorkonto med tilhørende fordring. Når dette fungerer, har du gode forudsætninger for implementering af de resterende funktionaliteter i operationen *OpretDebitorRegistrering*, hvilket er beskrevet i de efterfølgende afsnit.

Note: Hvis Debitorkonto *JuridiskAnsvarlig* ikke er et CPR- eller CVR-nummer (*lokal partidentifikation*), da skal du også angive den autoritære kilde for partidentifikation ved:

- DebitorkontoFelter
 - JuridiskAnsvarlig
 - OprettetAfStruktur
 - VirksomhedCVRNummer
 - ItSystemInstansIdentifikation
- JuridiskAnsvarligPartAdresse

3.6.2. DebitorkontoFelter

Benyttes kun når en ny konto skal oprettes, ellers udelades dette element. Kan således ikke efterfølgende opdateres. Eksempelværdier for de centrale elementer:

- DebitorkontoFelter
 - DebitorkontoType: 591f8439-6bae-49a2-9928-7b0f97831e90 (Havn)
 - DebitorkontoOrganisatoriskReference: 4053d58d-bd5e-48d9-b9d8-2c041faeccd5 (Team 1)
 - DebitorkontoIdentifikation: 6201670000
 - DebitorkontoIdentifikationType: da335bc8-28bd-408f-b7b2-a10a03a0f8f3
 - DebitorkontoAdministratorMarkering: true
 - DebitorkontoFølsomhed: 292e85a9-8ad4-46df-9e50-f97d6837ad74
 - JuridiskAnsvarlig
 - ØIRPartStruktur
 - ØIRPartId: 6201670000
 - ØIRPartIdType: 41a31ac1-27c0-4ef1-be80-175f7dd0de65

Kommentarer til udfyldte værdier:

Debitorkontoidentifikation + DebitorkontoidentifikationType

Til denne test - et fiktivt personnummer: 6201670000. Værdi fra HavneAdm sortiment for *DebitorkontoidentifikationType* "CPR-nummer".

DebitorkontoAdministratorMarkering

Værdi "true" angiver, at vi (fagsystemet) *har ansvaret for at vedligeholde en eventuel administrator for debitorkontoen.*

DebitorkontoFølsomhed

Værdi fra HavneAdm sortiment for *DebitorkontoFølsomhed* "Almindelige personoplysninger eller fortrolige forretningsdata"

ØIRPartId + ØIRPartIdType

Den juridisk ansvarlige for fordringen (den der skal betale). I dette eksempel samme person som angivet i *Debitorkontoidentifikation*. Bemærk, at både *DebitorkontoidentifikationType* og *ØIRPartIdType* begge angiver at Identifikation/PartId er et CPR-nummer - men det er forskellige værdier fra to forskellige værdilister.

Følgende ikke-påkrævede elementer er udeladt af eksempel:

DebitorkontoSagsreference
DebitorkontoSpecifikationListe
JuridiskAnsvarlig/OprettetAfStruktur

3.6.3.FordringListe

Eksempelværdier for de centrale elementer:

- FordringListe
 - Fordring
 - FordringUnikIdentifikation: 7cd86df8-306b-407e-ae16-7a3901554e5d
 - FordringKravType: 496dee04-de3a-4d77-bf17-c47de23b2a47 (Havneafgifter)
 - FordringOpkrævningTilstand: Opkraev
 - FordringFølsomhed: 292e85a9-8ad4-46df-9e50-f97d6837ad74
 - FordringTekstLinjeListe
 - FordringSpecifikationListe
 - FordringSpecifikationType: 719ba471-5394-42ad-898d-89c920b1d0c2
 - FordringSpecifikationTekstEllerKlasseValg
 - FordringSpecifikationTekstVærdi: Noget tekst
 - FordringSpecifikationType: b0598db0-5b2a-4112-bc84-f20219e96861
 - FordringSpecifikationTekstEllerKlasseValg
 - FordringSpecifikationKlasseVærdi: 83032492-33c6-4521-880e-7dbb2cfc15a8

Kommentarer til udfyldte værdier:

FordringUnikIdentifikation

Alle fordringer på listen betragtes som nye og skal have unikt UUID, der ikke tidligere er benyttet.

FordringKravType

Den første af de to mulige under DebitorkontoType *Havn*.

FordringOpkrævningTilstand

Husk at læse i servicebeskrivelsen de mulige værdier og i hvilke sammenhænge de skal benyttes:

FordringOpkrævningTilstand	<i>FordringOpkrævningTilstand</i> : String enumeration=Opkraev;IngenOpkraevning;AfvendtOpkraevning	Angiver, hvordan debitorsystemet skal håndtere <i>Fordringen</i> i forhold til opkrævning. "Opkraev": Debtorsystemet kan igangsætte opkrævning. Når denne værdi er angivet, må den ikke efterfølgende ændres. "IngenOpkraevning": Debtorsystemet må ikke foretage opkrævning af fordringen. Anvendes, når fordringen er oprettet af et fagsy-
----------------------------	---	--

FordringSpecifikationListe

Som vi så i afsnit om sortiment, er der en regel på Debitorkontotype *Havn*, der siger, at vi altid skal angive begge FordringSpecifikationTyper *Indrivelsestekst* og *PSRM kravtype*. Førstnævnte angives med tekstværdi og sidstnævnte er her angivet som eksempel værdien for PSRM Kravtype *Havneydelser*.

Følgende ikke-påkrævede elementer er udeladt af eksempel:

FordringEnhedsoplysning
DebitorKontaktinformation
BilagListe

3.6.4. Øvrige elementer

Følgende under-elementer til *OpretDebitorKontoRegistrering_I* er ikke inkluderet i eksempel og du bedes se servicebeskrivelsen for anvendelse og implementering:

- JuridiskAnsvarligPartAdresse
- Administrator
- JuridiskMedhæfterListe
- HæftelseListe
- FordringNedreguleringListe
- OpkrævningAnmodningListe (1)
- OpkrævningListe

(1) Benyttes udelukkende hvis man ønsker, at samle en række fordringer i samme opkrævning.

3.6.5. Eksempel

Her samlet eksempel af oprettelse af debitorKonto og tilhørende fordring. Der kaldes her med token-sikkerhed og derfor er *HovedOplysninger/AuthorityContext* udeladt.

OpretDebitorKontoRegistrering eksempel del #1

```
<OpretDebitorKontoRegistrering_I>
<ns1:HovedOplysninger xmlns:ns1="http://kombit.dk/xml/schemas/kontekst/2017/01/01/">
  <ns1:TransaktionsId>9ad2e646-2a9e-4332-86b8-18ccaf153aff</ns1:TransaktionsId>
  <ns1:TransaktionsTid>2020-01-28T12:00:23.922+01:00</ns1:TransaktionsTid>
  <ns1:Rute>
    <ns1:AfsenderOrganisation>11111111</ns1:AfsenderOrganisation>
    <ns1:AfsenderItSystemInstans>db7e15db-1ee3-48dc-2492-206dca18611f</ns1:AfsenderItSystemInstans>
    <ns1:ModtagerOrganisation>11111111</ns1:ModtagerOrganisation>
  </ns1:Rute>
</ns1:HovedOplysninger>
```

OpretDebitorKontoRegistrering eksempel del #2

```
<DebitorKontoUnikIdentifikation>41d54894-a515-4b75-8754-bbab3d756ff7</DebitorKontoUnikIdentifikation>
<DebitorKontoFelter>
  <DebitorKontoType>591f8439-6bae-49a2-9928-7b0f97831e90</DebitorKontoType>
```

```
<DebitorkontoOrganisatoriskReference>4053d58d-bd5e-48d9-b9d8-2c041faeccd5</DebitorkontoOrganisatoriskReference>
<DebitorkontoOprettetDatoTid>2022-09-28T12:00:23.922+01:00</DebitorkontoOprettetDatoTid>
<DebitorkontoIdentifikation>6201670000</DebitorkontoIdentifikation>
<DebitorkontoIdentifikationType>da335bc8-28bd-408f-b7b2-a10a03a0f8f3</DebitorkontoIdentifikationType>
<DebitorkontoAdministratorMarkering>true</DebitorkontoAdministratorMarkering>
<DebitorkontoFølsomhed>292e85a9-8ad4-46df-9e50-f97d6837ad74</DebitorkontoFølsomhed>
<JuridiskAnsvarlig>
  <ØIRPartStruktur>
    <ØIRPartId>6201670000</ØIRPartId>
    <ØIRPartIdType>41a31ac1-27c0-4ef1-be80-175f7dd0de65</ØIRPartIdType>
  </ØIRPartStruktur>
</JuridiskAnsvarlig>
</DebitorkontoFelter>
```

OpretDebitorkontoRegistrering eksempel del #3

```
<FordingListe>
  <Fording>
    <FordingUnikIdentifikation>7cd86df8-306b-407e-ae16-
7a3901554e5d</FordingUnikIdentifikation>
    <FordingOprettetDatoTid>2022-09-28T12:00:23.922+01:00</FordingOprettetDatoTid>
    <FordingKravType>496dee04-de3a-4d77-bf17-c47de23b2a47</FordingKravType>
    <FordingForfaldsdato>2022-11-29</FordingForfaldsdato>
    <FordingOpkrævningTilstand>Opkraev</FordingOpkrævningTilstand>
    <FordingBeløb>2000</FordingBeløb>
    <FordingStiftelsesdato>2022-09-28</FordingStiftelsesdato>
    <Opgørelsesperiode>
      <FordingPeriodeFraOgMedDato>2022-06-01</FordingPeriodeFraOgMedDato>
      <FordingPeriodeTilOgMedDato>2022-08-31</FordingPeriodeTilOgMedDato>
    </Opgørelsesperiode>
    <FordingFølsomhed>292e85a9-8ad4-46df-9e50-f97d6837ad74</FordingFølsomhed>
    <FordingTekstLinjeListe>
      <FordingTekstLinje>
        <FordingTekstLinjeIndex>1</FordingTekstLinjeIndex>
        <FordingTekstLinjeTekst>Tekstlinie 1</FordingTekstLinjeTekst>
      </FordingTekstLinje>
    </FordingTekstLinjeListe>
    <FordingSpecifikationListe>
      <FordingSpecifikation>
        <FordingSpecifikationType>92ef59ae-fa87-45d8-a6c0-
88f9f26c58e9</FordingSpecifikationType>
        <FordingSpecifikationTekstEllerKlasseValg>
          <FordingSpecifikationKlasseVærdi>83032492-33c6-4521-880e-
7dbb2cfc15a8</FordingSpecifikationKlasseVærdi>
        </FordingSpecifikationTekstEllerKlasseValg>
      </FordingSpecifikation>
      <FordingSpecifikation>
        <FordingSpecifikationType>719ba471-5394-42ad-898d-
89c920b1d0c2</FordingSpecifikationType>
        <FordingSpecifikationTekstEllerKlasseValg>
          <FordingSpecifikationTekstVærdi>Opkrævning af leje for bådplads MS Martha af
Svendborg</FordingSpecifikationTekstVærdi>
        </FordingSpecifikationTekstEllerKlasseValg>
      </FordingSpecifikation>
    </FordingSpecifikationListe>
  </Fording>
</FordingListe>
</OpretDebitorkontoRegistrering_I>
```

Før at eksempel virker for dig skal du:

- Erstatte værdier i Hovedoplysninger med din kommune og dit IT-system
- Erstatte klassificerede elementer med værdier fra dit sortiment

Have serviceaftaler godkendt for pågældende kommune

Når dette virker, og du har oprettet den første konto med tilhørende fordring i debitorsystemet, da er du klar til at implementere de resterende funktionaliteter i operation *OpretDebitorkontoRegistrering*.

3.7. Operation SætFordringTilOpkrævning

For at teste operationen starter vi med at kalde operation *OpretDebitorRegistrering*, for at oprette en ny fordring, der afventer opkrævning:

- FordringListe
 - Fordring
 - FordringUnikIdentifikation: 52aafe86-b665-4c8b-841c-f2d012c46add
 - FordringOpkrævningTilstand: AfventOpkraevning

Dernæst kan vi kalde denne operation, for at sætte fordring til opkrævning (namespace deklARATIONER udeladt):

```
<SætFordringTilOpkrævning_I>
  <HovedOplysninger>
    ...
  </HovedOplysninger>
  <DebitorkontoUnikIdentifikation>41d54894-a515-4b75-8754-
bbab3d756ff7</DebitorkontoUnikIdentifikation>
  <FordringListe>
    <Fordring>
      <FordringUnikIdentifikation>52aafe86-b665-4c8b-841c-f2d012c46add
</FordringUnikIdentifikation>
    </Fordring>
  </FordringListe>
  <FordringOpkrævningFraDato>2020-03-28</FordringOpkrævningFraDato>
</SætFordringTilOpkrævning_I>
```

Kommentarer til udfyldte værdier:

DebitorkontoUnikIdentifikation - ID for debitorkonto vi oprettede i forrige kapitel

FordringUnikIdentifikation - ID for fordring vi lige har oprettet til denne test

FordringOpkrævningFraDato - sæt til en hverdag i nærmeste fremtid

3.8. Operation AfslutHæftelse

For at kunne teste denne operation, starter vi med at oprette en hæftelse på den fordring vi oprettede tidligere. Vi kalder operation *OpretDebitorkontoRegistrering* med *HæftelseListe*:

- DebitorkontoUnikIdentifikation: 41d54894-a515-4b75-8754-bbab3d756ff7
- HæftelseListe
 - Hæftelse
 - HæftelseUnikIdentifikation: 995ec0a6-1199-4ccc-8109-54dc8dd4a034

- HæftelseOprettetDatoTid: 2020-01-29T12:00:23.922+01:00
- ØIRPartStruktur
 - ØIRPartId: 6301680000
 - ØIRPartIdType: 41a31ac1-27c0-4ef1-be80-175f7dd0de65
- HæfterForFordring
 - FordringUnikIdentifikation: 7cd86df8-306b-407e-ae16-7a3901554e5d
- HæftelsePeriodeFraOgMedDato: 2022-09-30

Vi har da oprettet en hæftelse på personnummer 6301680000.

Vi kan nu teste kald til operation *AfslutHæftelse* (namespace deklARATIONER udeladt):

```
<AfslutHæftelse_I>
  <HovedOplysninger>
    ...
  </HovedOplysninger>
  <DebitorKontoUnikIdentifikation>41d54894-a515-4b75-8754-
bbab3d756ff7</DebitorKontoUnikIdentifikation>
  <HæftelseUnikIdentifikation>995ec0a6-1199-4ccc-8109-
54dc8dd4a034</HæftelseUnikIdentifikation>
  <HæftelsePeriodeTilOgMedDato>2020-03-30</HæftelsePeriodeTilOgMedDato>
</AfslutHæftelse_I>
```

Kommentarer til udfyldte værdier:

- DebitorKontoUnikIdentifikation - ID for debitorKonto vi oprettede i forrige kapitel
- HæftelseUnikIdentifikation - ID for hæftelse vi lige har oprettet
- HæftelsePeriodeTilOgMedDato - en dato i fremtiden og senere end *HæftelsePeriodeFraOgMedDato*.

3.9. Operation ForetagEjerskifte

Som beskrevet i Servicebeskrivelsen, skal vi inden kald til denne operation:

1. Oprette en modtagende debitorKonto - dvs. en ny debitorKonto med juridisk ansvarlig, som fordringer skal overdrages til.
2. Oprette de aktuelle fordringer (restgælden) på den modtagende debitorKonto.
3. Nedregulere fordringer på afgivende debitorKonto, der overdrages, således at saldo ender på nul.

Alle tre trin udføres med kald til operation *OpretDebitorKontoRegistrering*. Vi skal huske at sætte *EjerskifteMarkering* på alle kald.

Hvis afgivende konto har en åben fordring på 2000 kr og en 500 kr nedregulering på fordringen, da kan man på modtagende konto oprette fordring på 1500 kr uden nedregulering.

Dernæst kan vi kalde operation *ForetagEjerskifte* (namespace deklARATIONER udeladt):

```

<ForetagEjerskifte_I>
  <HovedOplysninger>
    ...
  </HovedOplysninger>
  <DebitorkontoUnikIdentifikation>41d54894-a515-4b75-8754-
bbab3d756ff7</DebitorkontoUnikIdentifikation>
  <EjerskiftesTilDebitorkonto>
    <DebitorkontoUnikIdentifikation>8ed23c89-f8ef-45e2-a887-
11a4aaf76907</DebitorkontoUnikIdentifikation>
  </EjerskiftesTilDebitorkonto>
</ForetagEjerskifte_I>

```

Vi angiver blot ID for afgivende og modtagende debitor-konto.

3.10. Operation AfslutAdministrator

For at kunne teste denne operation opretter vi først en Administrator på Debitorkonto. Dette gør vi ved at kalde operation *OpretDebitorkontoRegistrering* med ID for Debitorkonto og element Administrator:

- DebitorkontoUnikIdentifikation: 41d54894-a515-4b75-8754-bbab3d756ff7
 - Administrator
 - ØIRPartStruktur
 - ØIRPartId: 11223344
 - ØIRPartIdType: 425a09c9-7ab6-4fe9-83d2-88ab454bddaf

Kommentarer til anvendte værdier:

- DebitorkontoUnikIdentifikation - konto vi oprettede i forrige kapitel
- ØIRPartId - et fiktivt CVR-nummer, kunne fx være advokatbureau
- ØIRPartIdType - værdi for CVR-nummer i HavneAdm Debtorsortiment

Følgende valgfri elementer til Administrator er ikke angivet i dette eksempel:

OprettetAfStruktur (hvis denne er angivet under oprettelse af administrator, da skal den inkluderes i input til *AfslutAdministrator*)

ØIRPartAdresseStruktur

ProduktionsenhedPNummerStruktur

Når vi har tilføjet pågældende administrator til debitor-konto, da kan nu teste at afslutte denne ved kald til operation

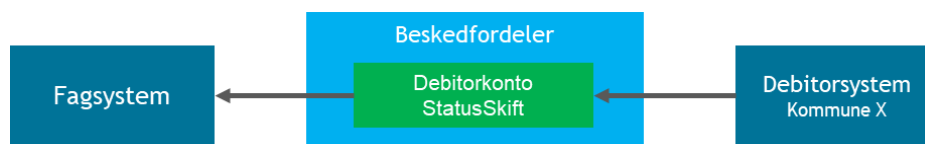
```

<AfslutAdministrator_I>
  <HovedOplysninger>
    ...
  </HovedOplysninger>
  <DebitorkontoUnikIdentifikation>41d54894-a515-4b75-8754-
bbab3d756ff7</DebitorkontoUnikIdentifikation>
  <ØIRPartStruktur>

```

```
<ØIRPartId>11223344</ØIRPartId>  
<ØIRPartIdType>425a09c9-7ab6-4fe9-83d2-88ab454bddaf</ØIRPartIdType>  
</ØIRPartStruktur>  
</AfslutAdministrator_I>
```

4. SERVICE DEBITORKONTOSTATUSSKIFT



Vi modtager hændelser fra debitorsystemet via Beskedfordeler.

4.1. Dokumentation

De centrale dokumenter i dokumentationspakken:

Besked service DebitorkontoStatusSkiftHent_1.0.zip

[SB] Servicebeskrivelse til DebitorkontoStatusSkiftHent_1.0.pdf

Bilag til SF1590_B_01 Integrationsbeskrivelse til Afsend DebitorRegistrering_3.9.zip

[BE] Beskrivelse til beskedtype DebitorkontoStatusSkift_1.0.pdf

[IM] Informationsmodel for dataudveksling med Debitor-domænet_1.5.pdf

4.2. Konfiguration

Vi anmoder om serviceaftale til Beskedfordeler på beskedtype DebitorkontoStatusSkift med UUID:

- 2e844a0f-6e13-4d3a-971c-0ad59b24898d

I ExtTest anmoder vi også om serviceaftale til at kunne sende beskeder – det skal du ikke gøre i Produktion, her er det debitorsystemet, der afsender beskederne. Vi har her benyttet Korsbæk kommune (11111111) - husk at angive kommune du tester med.

Bekræft serviceaftaleanmodning for it-system:

- KDI CTT Test System #2

Med anmodning om tilslutning til følgende services:

- afsend

Type: Fælleskommunal Service
Udbyder: Beskedfordeleren
Roller: Aflever

- BeskedType: 2e844a0f-6e13-4d3a-971c-0ad59b24898d
- Foelsomhed: 292e85a9-8ad4-46df-9e50-f97d6837ad74
- Kommunalt_forvaltningsomraade: *
- Tilladte Modtagere: 11111111

- afhent

Type: Fælleskommunal Service
Udbyder: Beskedfordeleren
Roller: Modtag

- Afsendende_myndighed: 11111111
- BeskedType: 2e844a0f-6e13-4d3a-971c-0ad59b24898d
- Foelsomhed: 292e85a9-8ad4-46df-9e50-f97d6837ad74
- Kommunalt_forvaltningsomraade: *

Send anmodninger til følgende dataejer(e):**Send anmodninger til følgende datamodtager(e):**

- Korsbaek Kommune, CVR nummer 11111111

Dernæst opretter vi et dueslag og tilføjer et abonnement. Du kan også tilføje abonnement til et eksisterende dueslag, alt efter smag og behag. Vi tilføjer et abonnementsudtryk, så vi er sikre på, kun at modtage hændelser relateret til konti vi har oprettet.

Rediger abonnementsudtryk

*Abonnements udtryks ident:

*Navn:

*Aktiv: Ja Nej

*Beskedtype:

Værdiliste:

*Gyldig fra:

dd/mm/åååå tt:mm:ss

Gyldig til:

dd/mm/åååå tt:mm:ss

*Modtag egne beskeder: Ja Nej

Udtryk:

```
BeskedVersion = '1.0' AND  
get(Beskedkuvert.Filtreringsdata.RelateretObjekt,2).ObjektId.UUIDIdentifikator = 'f269ca0c-7b5d-49e2-a585-1ee39ba5a431' AND  
get(Beskedkuvert.Filtreringsdata.RelateretObjekt,2).ObjektType.UUIDIdentifikator = '91aa31b3-c4b3-499e-8689-93b1ef89f319' AND  
get(Beskedkuvert.Filtreringsdata.RelateretObjekt,2).ObjektRolle.UUIDIdentifikator = '07c8ac8b-2173-45bf-a6ad-332bcf018a32'
```

Vær opmærksom på, hvis du kopierer udtryk fra PDF-vejledningen:

Der kan være linjeskift efter anførselstegn du skal fjerne
Anførselstegn kan være i kursiv og skal ændres til lige anførselstegn

4.3. Eksempelbesked

Da modtagere af hændelse *DebitorkontoStatusSkift* skal oprette abonnementsudtryk, der sikrer, at de kun modtager kopi af hændelser relateret til konti, de selv har oprettet, da kan vi sende testbeskeder til os selv, uden at forstyrre andre. Også grundet, at de fleste kommuner ikke har etableret en test-instans af deres økonomisystem, så vi kan kun modtage reelle hændelser i produktionsmiljøet.

For at kunne teste denne service, etablerer vi således en testbesked i henhold til [BE] *Beskrivelse til beskedtype DebitorKontoStatusSkift_1.0.pdf* - afsnit *Elementbeskrivelse*. Værdier markeret med gult skal erstattes med egne værdier.

Komplet hændelsesbesked del #1

```
<ns2:Haendelsesbesked xmlns:ns1="urn:oio:sagdok:3.0.0"
xmlns:ns2="urn:oio:besked:kuvert:1.0">
  <ns2:BeskedId>
    <ns1:UUIdIdentifikator>2ea77a77-77a2-49ac-883f-bdc9d30da2fa</ns1:UUIdIdentifikator>
  </ns2:BeskedId>
  <ns2:BeskedVersion>1.0</ns2:BeskedVersion>
  <ns2:Beskedkuvert>
    <ns2:Filtreringsdata>
      <ns2:Beskedtype>
        <ns1:UUIdIdentifikator>2e844a0f-6e13-4d3a-971c-
0ad59b24898d</ns1:UUIdIdentifikator>
      </ns2:Beskedtype>
      <ns2:BeskedAnsvarligAktoer>
        <ns1:UUIdIdentifikator>f269ca0c-7b5d-49e2-a585-
1ee39ba5a431</ns1:UUIdIdentifikator>
      </ns2:BeskedAnsvarligAktoer>
      <ns2:TilladtModtager>
        <ns1:URNIdentifikator>urn:oio:cvr-nr:11111111</ns1:URNIdentifikator>
      </ns2:TilladtModtager>
    </ns2:Filtreringsdata>
  </ns2:Beskedkuvert>
</ns2:Haendelsesbesked>
```

1. Angiv nyt BeskedId hver gang du sender.
2. UUID for Debtorsystemet. Da vi selv sender beskeden i testmiljøet, som var vi et debtorsystem, da angiver vi vores eget anvendersystem - erstat med værdi for dit system.
3. Erstat med CVR-nummer for kommune du tester med.

Komplet hændelsesbesked del #2

```
<ns2:RelateretObjekt>
  <ns2:ObjektId>
    <ns1:URNIdentifikator>2201670000</ns1:URNIdentifikator>
  </ns2:ObjektId>
  <ns2:ObjektType>
    <ns1:UUIdIdentifikator>da335bc8-28bd-408f-b7b2-
a10a03a0f8f3</ns1:UUIdIdentifikator>
  </ns2:ObjektType>
  <ns2:ObjektRolle>
    <ns1:UUIdIdentifikator>1df113f4-adb7-4c6f-aaa3-
82eadaad63fc</ns1:UUIdIdentifikator>
  </ns2:ObjektRolle>
</ns2:RelateretObjekt>
<ns2:RelateretObjekt>
  <ns2:ObjektId>
    <ns1:URNIdentifikator>2201670000</ns1:URNIdentifikator>
  </ns2:ObjektId>
  <ns2:ObjektType>
```

```

    <ns1:UUIDIdentifikator>41a31ac1-27c0-4ef1-be80-
175f7dd0de65</ns1:UUIDIdentifikator>
    </ns2:ObjektType>
    <ns2:ObjektRolle>
      <ns1:UUIDIdentifikator>12a391db-95d3-41d2-8738-
92047dd5b953</ns1:UUIDIdentifikator>
    </ns2:ObjektRolle>
  </ns2:RelateretObjekt>
  <ns2:RelateretObjekt>
    <ns2:ObjektId>
      <ns1:UUIDIdentifikator>f269ca0c-7b5d-49e2-a585-
1ee39ba5a431</ns1:UUIDIdentifikator>
    </ns2:ObjektId>
    <ns2:ObjektType>
      <ns1:UUIDIdentifikator>91aa31b3-c4b3-499e-8689-
93b1ef89f319</ns1:UUIDIdentifikator>
    </ns2:ObjektType>
    <ns2:ObjektRolle>
      <ns1:UUIDIdentifikator>07c8ac8b-2173-45bf-a6ad-
332bcf018a32</ns1:UUIDIdentifikator>
    </ns2:ObjektRolle>
  </ns2:RelateretObjekt>

```

1. RelateretObjekt - Debitorkontoidentifikation
2. Debitor CPR-nummer vi tidligere oprettede en konto for.
3. DebitorkontoidentifikationType *CPR-nummer* - erstat med værdi fra eget Sortiment.
4. RelateretObjekt - JuridiskAnsvarlig
5. Vi har på konto oprettet tidligere samme person som juridisk ansvarlig.
6. ØIRPartIdType *CPR-nummer* - erstat med værdi fra eget Sortiment.
7. RelateretObjekt - system der har oprettet eller overtaget debitorkontoen, erstat med værdi for dit IT-system

Komplet hændelsesbesked del #3

```

  <ns2:ObjektRegistrering>
    <ns2:ObjektRegistreringId>
      <ns1:UUIDIdentifikator>b2efe672-ac53-4c79-9c68-
9871d71a5577</ns1:UUIDIdentifikator>
    </ns2:ObjektRegistreringId>
    <ns2:RegistreringsAktoer>
      <ns1:UUIDIdentifikator>f269ca0c-7b5d-49e2-a585-
1ee39ba5a431</ns1:UUIDIdentifikator>
    </ns2:RegistreringsAktoer>
    <ns2:Registreringstidspunkt>
      <ns1:TidsstempelDatoTid>2022-08-17T08:00:00</ns1:TidsstempelDatoTid>
    </ns2:Registreringstidspunkt>
    <ns2:ObjektAnsvarligMyndighed>
      <ns1:URNIdentifikator>urn:oio:cvr-nr:11111111</ns1:URNIdentifikator>
    </ns2:ObjektAnsvarligMyndighed>

```

```

        <ns2:ObjektId>
            <ns1:UUIDIdentifikator>41d54894-a515-4b75-8754-
bbab3d756ff7</ns1:UUIDIdentifikator>
        </ns2:ObjektId>
        <ns2:ObjektType>
            <ns1:UUIDIdentifikator>0fbd1706-71b2-4517-b20b-
57dead080a2a</ns1:UUIDIdentifikator>
        </ns2:ObjektType>
        <ns2:ObjektHandling>
            <ns1:UUIDIdentifikator>3642d7d0-059e-45fb-9279-
2fc0c11d9e4d</ns1:UUIDIdentifikator>
        </ns2:ObjektHandling>
        <ns2:OpgaveEmne>
            <ns1:URNIdentifikator>*</ns1:URNIdentifikator>
        </ns2:OpgaveEmne>
    </ns2:ObjektRegistrering>
</ns2:Filtreringsdata>
    
```

1. Unik identifikation af den hændelse, som er grundlaget for besked forsendelsen dannes af debitorsystemet. Revisionsspor.
2. System der sendte beskeden - erstat med dit system. Vil i produktion være UUID for debitorsystemet.
3. Blot husker om at sætte fornuftig dato/tid.
4. Skift CVR-nummer til kommune du tester med.
5. UUID for debitorkonto vi oprettede tidligere - erstat med egen værdi.
6. Værdi for handling DebitorkontoAfsluttet.

Komplet hændelsesbesked del #4

```

<ns2:Leveranceinformation>
    <ns2:Dannelsestidspunkt>
        <ns1:TidsstempelDatoTid>2022-08-17T09:00:56.547+02:00</ns1:TidsstempelDatoTid>
    </ns2:Dannelsestidspunkt>
    <ns2:Sikkerhedsklassificering>
        <ns1:UUIDIdentifikator>292e85a9-8ad4-46df-9e50-
f97d6837ad74</ns1:UUIDIdentifikator>
    </ns2:Sikkerhedsklassificering>
</ns2:Leveranceinformation>
<ns2:GyldighedFra>
    <ns1:TidsstempelDatoTid>2022-08-17T08:00:00</ns1:TidsstempelDatoTid>
</ns2:GyldighedFra>
<ns2:GyldighedTil>
    <ns1:TidsstempelDatoTid>2022-08-17T08:00:00</ns1:TidsstempelDatoTid>
</ns2:GyldighedTil>
</ns2:Beskedkuvert>
<ns2:Beskeddata/>
</ns2:Haendelsesbesked>
    
```

1. Blot husker om at sætte fornuftig dato/tid.
2. GyldighedFra - Dato hvorfra statusskiftet er aktuelt. Dvs. første dag i virkningsperioden.

3. GyldighedTil

- a. **DebitorkontoAfsluttet:** Sættes til sidste dag i perioden, hvor debitorkontoen har tilstanden afsluttet. Herefter vil debitorkontoen skifte til tilstanden kasseret. Kan her godt være lig GyldigFra.
- b. **DebitokontoGenåbnet:** Her angives ikke nogen dato for tilstandens afslutning.

Vi har nu en komplet hændelsesbesked, der notificerer om at debitorkonto, vi oprettede i starten af vejledningen, er blevet afsluttet. Inden vi sender beskeden kan vi teste, at abonnementsudtryk fungerer. Man vælger blot "Afprøv udtryk" og man dernæst vælge fil med besked, man ønsker at teste mod udtrykket. Nederst i formularen vælger du "Valider mod besked".

Udtryk:

```
BeskedVersion = '1.0' AND  
get(Beskedkuvert.Filtreringsdata.RelateretObjekt,2).ObjektId.UUIDIdentifikator = 'f269ca0c-7b5d-49e2-a585-1ee39ba5a431' AND  
get(Beskedkuvert.Filtreringsdata.RelateretObjekt,2).ObjektType.UUIDIdentifikator = '91aa31b3-c4b3-499e-8689-93b1ef89f319' AND  
get(Beskedkuvert.Filtreringsdata.RelateretObjekt,2).ObjektRolle.UUIDIdentifikator = '07c8ac8b-2173-45bf-a6ad-332bcf018a32'
```

Slet Historik Afprøv udtryk Valider udtryk Annuller Gem

Vælg besked fil

Succes

Besked xml:

```
<ns2:Haendelsesbesked xmlns:ns1="urn:oio:sagdok:3.0.0"  
xmlns:ns2="urn:oio:besked:kuvert:1.0">  
  <ns2:BeskedId>  
    <ns1:UUIDIdentifikator>2ea77a77-77a2-49ac-883f-  
bdc9d30da2fa</ns1:UUIDIdentifikator>  
  </ns2:BeskedId>  
  <ns2:BeskedVersion>1.0</ns2:BeskedVersion>  
  <ns2:Beskedkuvert>  
    <ns2:Filtreringsdata>  
      <ns2:Beskedtype>  
        <ns1:UUIDIdentifikator>2e844a0f-
```

Valider mod besked Annuller

Det er beskrevet i Kom-godt-i-gang *Beskedfordeler*, hvorledes du sender en besked med Java eksempelkoden og med .net. Vi har nu sendt beskeden:

```
...  
- main: Sending message...  
- publishMessage: Publishing message with transactionId=fa41da50-ace2-457b-a241-7c265b2b9ca9
```

```

- Received reply for request with correlation id 8f7e7a3a-46a7-44d6-8752-3e6665d9d76a
- Received reply:
***
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<StandardRetur xmlns="urn:oio:sagdok:3.0.0" xmlns:ns2="urn:oio:besked:kuvert:1.0"
xmlns:ns3="http://www.w3.org/2000/09/xmldsig#">
  <StatusKode>20</StatusKode>
  <FejlbeskedTekst>OK</FejlbeskedTekst>
</StandardRetur>

***
- Publish executed successfully
- Message sent successfully.
- main: Closing connection...

```

Beskeden er modtaget i vores dueslag:

Dueslags indhold

Vælg type Søg efter BeskedId

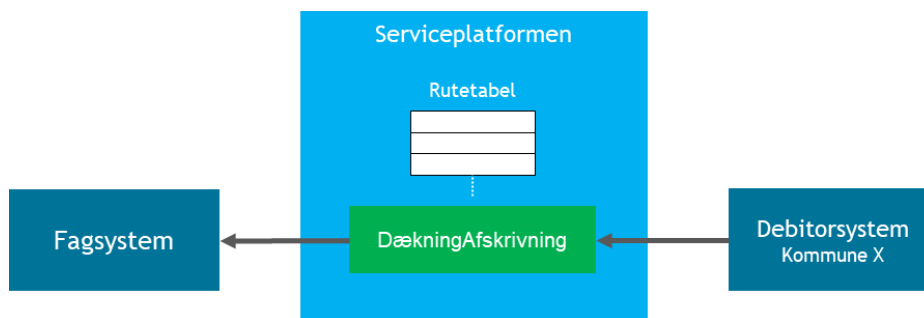
Vælg alle	BeskedId	Beskedtype	Kildesystem	Dannelsestidspunkt
<input type="checkbox"/>	5bea048d-c404-46db-bcce-8fd71eb53ddb	OekonomiDebitor Konto Status S kift	KDI CTT Test System #2	17/08/2022 10:11:30

Java-klient vi benyttede kan sættes til automatisk at sætte et nyt BeskedId, hvilket vi har gjort her. Hvis du skal teste med at sende en *DebitorKontoGenaktiveret* hændelse, husk at:

ObjektHandling skal sættes til 50c77c76-3627-412e-bca2-72a6a6c8b39e

GyldighedTil element skal fjernes

5. SERVICE DÆKNINGAFSKRIVNING



Vi modtager dækninger og afskrivninger fra debitorsystemet, på debitor konti vi har oprettet, via en webservice vi udstiller.

Debitorsystemet kan notificere os (fagsystemet), når betaling er sket (dækning) eller en fordring reduceres (afskrivning) fx ved gældssanering. Servicenavnet skal således forstås som, at der kan registreres både dækninger og afskrivninger på fordringer (ikke "Afskrivning af dækning"). Integrationen skal aftales mellem parterne.

Debitorsystemet kan nedregulere en dækning eller afskrivning helt eller delvist, fx ved annullering eller flytning til anden konto.

5.1. Dokumentation

Fra dokumentationspakken:

Web service DækningAfskrivningRegistreringAfsend_1.1.zip

- Servicebeskrivelse til DækningAfskrivningRegistreringAfsend_1.1.pdf
- WSDL og XSDs til DækningAfskrivningRegistreringAfsend_1.1.zip
- Aarsagskoder_DækningAfskrivningRegistreringAfsend_1.csv

5.2. Implementer sikkerhed

Inden du kan etablere og registrere din service, skal du vælge sikkerhedsmodel, du ønsker at benytte. Du har to muligheder:

- Simpel fælleskommunal = AuthorityContext (1)
 - Myndighed-kontekst er angivet i HovedOplysninger
 - Forespørgsel indeholder ikke XML signatur - svar skal ikke signeres
- Fælleskommunal = Security Token (2)
 - Myndighed-kontekst er angivet i SAML-token
 - Forespørgsel indeholder XML signatur - svar skal også signeres

(1,2) Begge sikkerhedsmodeller kræver, at du implementerer 2-vejs-TLS og validerer certifikat Serviceplatformen kalder med. Dette er illustreret med praktisk eksempel i [Kom-godt-i-gang Certifikater](#).

(1) Ved anvendelse af AuthorityContext aflæser og validerer du *HovedOplysninger* i forespørgsel.

(2) Ved anvendelse af Fælleskommunal sikkerhedsmodel skal du benytte et SAML-rammeverk, der understøtter token-sikkerhed og Liberty Basic SOAP binding. For detaljeret information, venligst se [Adgangsstyring for systemer](#). For praktiske detaljer vedrørende udstilling af webservice med token-sikkerhed, venligst se [Kom-godt-i-gang Udstil Webservice](#).

Husk at teste sikkerhed på din service, inden du anmoder om gennemstilling via Serviceplatformen. Lav en SOAP-klient og prøv at kalde service med eget funktionscertifikat i henhold til valgt sikkerhedsmodel.

5.3. Registrer service i FK ADM

På vores IT-system i FK ADM aktiverer vi typen "Serviceudbyder", hvorefter faneblad "Serviceudbyder" bliver aktivt:

Type:

- Anvendersystem
- Brugervendt system
- Identity Provider
- Serviceudbyder

Vær opmærksom på at du ikke kan vælge systemtyper fra, når først du har gemt.

Registrer servicesystemrolle

NB: Kun hvis du udestiller service med token-sikkerhed (Fælleskommunal Service). Servicesystemroller benyttes kun ved token-sikkerhed, så hvis du benytter AuthorityContext behøver du ikke oprette en sådan.

Klik på overskriften *Servicesystemroller*, da ser du listen og kan vælge at oprette:

Stamdata	Dataafgrænsningstyper	Anvendersystem	Brugervendt system	Identity Provider	Serviceudbyder
Servicesystemroller					
+ Opret servicesystemrolle					

Vi opretter en Servicesystemrolle:

Navn: Fx "Registrering" - du vælger selv et navn til rollen.

Domæne: Angiv domæne for dit fagsystem.

Rollenavn: Angiv samme som under Navn - dog ingen mellemrum eller specielle tegn.

Version: 1.

Dataafgrænsningstyper: Ingen.

Eksempel:

Opret servicesystemrolle

UUID: 777ef579-22ab-4f8c-e6fd-d402e0180584

Navn: *

Beskrivelse:

EntityId ?

Domæne: *

Rollenavn: *

Version: *

http://mit-fagsystem.dk/roles/servicesystemrole/registrering/1

Registrer service

Klik på "Services" for at udfolde denne sektion og vælg dernæst "Opret service":

KDI CTT Test System #2

Stamdata
Dataafgrænsningstyper
Anvendelsesystem
Brugervendt system
Identity Provider
Serviceudbyder

Servicesystemroller ▶

Services ▼

+ Opret service

Her med eksempelverdier:

Opret service

Stamdata	Servicesystemroller	Afgrensning af myndigheder
UUID: *	<input type="text" value="6c0a686d-3c9b-42d3-df4b-e791a9971b71"/>	
Navn: *	<input type="text" value="KOMBIT CTT SF1590 B 01 Dækning og Afskrivning registrering"/>	
EntityId ?		
Domæne: *	<input type="text" value="mit-fagsystem.dk"/>	
Servicenavn: *	<input type="text" value="DaekningAfskrivningRegistrering"/>	
Version: *	<input type="text" value="1"/>	
http://mit-fagsystem.dk/service/DaekningAfskrivningRegistrering/1		

Servicespecifikke vilkår:

B
I
H
🗨
☰
☰
🔗
👁
?

Denne service er alene tiltænkt debitorsystemer, der skal kunne registrere dækning og afskrivning på fordringer oprettet af vores fagsystem.

Under "Servicetype" vælger du sikkerhedsmodel du udstiller service med. Vælg:

- "Simpel Fælleskommunal Service", hvis du benytter AuthorityContext
- "Fælleskommunal Service", hvis du benytter token-sikkerhed

Servicetype: *	<input type="text" value="Fælleskommunal Service"/>
Link til dokumentation:	<input type="text"/>
Videregivelse:	<input type="checkbox"/> der videregives data
Præ-config-URL:	<input type="text"/> ?
Post-config-URL:	<input type="text"/> ?

Vælg "Næste" fornedet; da kommer du til "Servicesystemroller". Hvis du har valgt "Simpel Fælleskommunal Service", da anvendes ikke Servicesystemroller og du kan blot vælge "Næste" igen:

Opret service

Stamdata	Servicesystemroller	Afgrænsning af myndigheder
Navn:	KOMBIT CTT SF1590 B 01 Dækning og Afskrivning registrering	
Systemroller:	Servicesystemroller bruges kun i forbindelse med fælleskommunale services.	

Hvis du har valgt “Fælleskommunal Service”, da skal du vælge den Servicesystemrolle du oprettede:

Opret service

Stamdata **Servicesystemroller** Afgrænsning af myndigheder

Navn: KOMBIT CTT SF1590 B 01 Dækning og Afskrivning registrering

Systemroller: Registrering ✕

Vælg “Næste” igen, og da kommer du til afgrænsning af myndigheder:

Opret service

Stamdata Servicesystemroller **Afgrænsning af myndigheder**

Navn: KOMBIT CTT SF1590 B 01 Dækning og Afskrivning registrering

Myndighed: Korsbæk Kommune ✕

Vælg alle Fravælg alle

Navn ^

Korsbæk Kommune

Forrige Annuller Færdig

Vælg relevante kommune/kommuner og dernæst “Færdig”.

Huske at gemme ændringer til IT-systemet til sidst. Nu er service registreret, og der kan anmodes om serviceaftale til denne.

Tip: I testøjemed og hvis du udstiller din service med token-sikkerhed; anmod om serviceaftale til din egen service. Da kan du teste, at det er muligt at trække et token, og du kan kalde din service selv direkte med eget klient-certifikat og udstedt token. Du kan dermed verificere, at du har korrekt implementeret SAML token-sikkerhed på din webservice, inden du bestiller gennemstilling fra Serviceplatformen.

5.4. Bestil gennemstilling på Serviceplatformen

Du skal bestille ydelsen [Oprettelse af serviceudbyders webservice](#). Du finder formularen nederst på siden. Bemærk forudsætning: "Servicen skal være sat i drift og funktionsduelig på den it-systeminstans, som udgør serviceudbydersystemet". Husk at:

- Først bestilles ydelsen og du modtager et bestillings ID
- Dernæst kan formularen sendes til Helpdesk
- Formularen skal indsendes fra e-mail, der er under Stamdata på IT-systemet i FK ADM (*)

(*) Du kan angive yderligere kontakt e-mails i kommentarfeltet under Stamdata.

Efter oprettelse af gennemstilling vil du blive informeret om, hvilket certifikat Serviceplatformen kalder med til etablering af gensidig TLS.

5.5. Operation Ping

For detaljer se afsnit *Integrationsflow: DækningAfskrivningRegistreringAfsend_v1_Ping_Fagsystem* i Servicebeskrivelsen. Når Serviceplatformen har identificeret sig med klient-certifikat og du har valideret dette, og der er etableret 2-vejs-TLS, da modtager du følgende forespørgsel:

Ping forespørgsel fra Serviceplatformen

(SOAP-elementer udeladt)

```
xmlns:ns="http://www.kombit.dk/2017/01/01/"
xmlns:ns1="http://kombit.dk/xml/schemas/kontekst/2017/01/01/"
..
<ns:Ping_I>
  <ns1:HovedOplysninger>
    <ns1:TransaktionsId>4ca3c91d-7d8f-42a6-a2bb-0d69ac5822c3</ns1:TransaktionsId>
    <ns1:TransaktionsTid>2022-05-10T12:24:00Z</ns1:TransaktionsTid>
    <ns1:AuthorityContext>
      <ns1:MunicipalityCVR>19435075</ns1:MunicipalityCVR>
    </ns1:AuthorityContext>
    <ns1:Rute>
      <ns1:AfsenderOrganisation>19435075</ns1:AfsenderOrganisation>
      <ns1:AfsenderItSystemInstans>19dc3a1d-c629-4853-b566-7fe49cf674ce</ns1:AfsenderItSystemInstans>
      <ns1:ModtagerOrganisation>19435075</ns1:ModtagerOrganisation>
```

```
<ns1:ModtagerItSystemInstans>f269ca0c-7b5d-49e2-a585-1ee39ba5a431</ns1:ModtagerItSystemInstans>
</ns1:Rute>
</ns1:HovedOplysninger>
</ns:Ping_I>
```

- Vi registrerer TransaktionsId og TransaktionsTid, så vi kan returnere disse i svaret
- Ved Ping kalder Serviceplatformen på vegne af KOMBIT (19435075)
- AuthorityContext er ikke med, hvis du har valgt at udstille service med token-sikkerhed
- AfsenderItSystemInstans er ID for Serviceplatformen - samme værdi i ExtTest og Produktion
- ModtagerItSystemInstans er ID for dit fagsystem

Mht. til svar vi skal returnere, du finder detaljer i [Webservice - Transaktionsspor og fejlbehandling ved servicekald](#) afsnit *HovedOplysningerSvar*. Udfør passende tjek af dit fagsystems helbred, fx at der er hul igennem til databasen. Hvis du udstiller service med token-sikkerhed, da skal du også her implementere validering af token, der kaldes med.

Når alt er OK svarer vi tilbage med HTTP 200:

Ping svar - HTTP 200 OK

(SOAP-elementer udeladt)

```
xmlns:ns="http://www.kombit.dk/2017/01/01/"
xmlns:ns1="http://kombit.dk/xml/schemas/kontekst/2017/01/01/"
...
<ns:Ping_0>
  <ns1:HovedOplysningerSvar>
    <ns1:TransaktionsId>4ca3c91d-7d8f-42a6-a2bb-0d69ac5822c3</ns1:TransaktionsId>
    <ns1:TransaktionsTid>2022-05-10T12:24:00Z</ns1:TransaktionsTid>
    <ns1:SvarReaktion>
      <ns1:Advis>
        <ns1:AdvisId>OK</ns1:AdvisId>
        <ns1:AdvisTekst>System is functional</ns1:AdvisTekst>
        <ns1:KildeId>f269ca0c-7b5d-49e2-a585-1ee39ba5a431</ns1:KildeId>
      </ns1:Advis>
    </ns1:SvarReaktion>
  </ns1:HovedOplysningerSvar>
</ns:Ping_0>
```

- Vi returnerer TransaktionsId og TransaktionsTid vi modtog.
- Vi har kun <Advis> under <SvarReaktion>, hvilket implicit indikerer, at der ingen fejl er.
- Vi definerer selv AdvisId - fx blot "OK". Ved succes er der ikke behov for andre returkoder.
- Under Kildeld angiver du ID for dit fagsystem.

Hvis din service ikke er funktionel, da skal du returnere HTTP 500 og angive fejlkode samt fejlårsag. Du finder liste af fejlkoder under servicedetaljer i [Digitaliseringskataloget](#) - "Link til fejlkode":

Webservice / Serviceplatformen

DækningAfskrivningRegistreringAfsend

Version
1.1

I DRIFT

✕

Eksternt testmiljø

Endpoint: <https://exttest.serviceplatformen.dk/service/OIR/DækningAfskrivningRegistreringAfsendService/1>

UUID: 67275f52-f9a2-484e-a869-92cad1a5572c

EntityId: <http://oir.serviceplatformen.dk/service/daekningafskrivningregistreringafsend/1>

Fejlkoder: [Link til fejlkode](#)

Her eksempel:

Ping svar - HTTP 500 Fejl

(SOAP-elementer udeladt)

```

xmlns:ns="http://www.kombit.dk/2017/01/01/"
xmlns:ns1="http://kombit.dk/xml/schemas/kontekst/2017/01/01/"
...
<ns:Ping_0>
  <ns1:HovedOplysningerSvar>
    <ns1:TransaktionsId>4ca3c91d-7d8f-42a6-a2bb-0d69ac5822c3</ns1:TransaktionsId>
    <ns1:TransaktionsTid>2022-05-10T12:24:00Z</ns1:TransaktionsTid>
    <ns1:SvarReaktion>
      <ns1:Fejl>
        <ns1:FejlId>ServiceSuspended</ns1:FejlId>
        <ns1:FejlTekst>Service is suspended</ns1:FejlTekst>
        <ns1:KildeId>f269ca0c-7b5d-49e2-a585-1ee39ba5a431</ns1:KildeId>
      </ns1:Fejl>
    </ns1:SvarReaktion>
  </ns1:HovedOplysningerSvar>
</ns:Ping_0>

```

- Her returnerer vi også modtaget TransaktionsId samt TransaktionsTid
- Vi angiver fejlkode og fejltekst
- Og angiver ID for vores IT-system som Kildeld

5.6. Operation OpretDækningAfskrivning

5.6.1. Validering

Se afsnit *Valideringsmodel* i servicebeskrivelsen. Det er vigtigt, at du læser hele kapitel grundigt, da der er mange regler, der skal implementeres. Validering er opdelt i:

- Generel leverancekontrol
- Forretningsmæssig validering

For sidstnævnte se afsnit *Valideringsregler* i kapitel *OpretDækningAfskrivning [Operation]* i servicebeskrivelsen. Bemærk også følgende vigtige note:

Den forretningsmæssige validering skal gennemføres for den fulde debitorregistrering. Det betyder, at selvom et element i debitorregistreringen fejler, så skal der fortsættes med validering af de efterfølgende elementer med mindre andet eksplicit er angivet under valideringsregler. Det betyder, at debitorsystemet via forretningsvaret får et fuldt overblik i, hvilke dele af dataleverancen, som er fejlbehæftet.

5.6.2. Generel leverancekontrol

Se afsnit *Generel leverancekontrol* i servicebeskrivelsen. Til at starte med, implementerer vi de påkrævede valideringer:

1. Skemavalidering af den samlede leverance
2. Sikring mod parallel håndtering af samme primære informationsobjekt
3. Kontrol af genfremsendelse

Servicebeskrivelsen indeholder detaljer om hvorledes hver regel udføres og håndteres, eksempel:

4.3.1 Skemavalidering

Formål	Formålet er at verificere om den samlede leverance kan tolkes syntaktisk.
Indgangsbetingelse	Ingen
Validering	Valideringen består af flg. trin 1. Skemavalidering af den samlede leverance
Positivt resultat	Der fortsættes til næste trin.
Negativt resultat	Servicen svarer med brug af HovedOplysningerSvar.
Note	Detailresultatet af skemavalidering logges til brug for senere debugging.

4.3.1.1 Fejlårsagskoder

Reference		FejlÅrsag		Supplerende angivelse af ObjektNavn ObjektUnikidentifikation
Id	Element i datastruktur	Kode	Titel	
1		01.0001.001	Leverancen kan ikke skemavalideres	

Sikring mod parallel håndtering af samme primære informationsobjekt

De primære informationsobjekter i *OpretDækningAfskrivning_I* er følgende, markeret med fed:

- FordringAfskrivningListe (0 eller 1)
 - FordringAfskrivning (0 eller flere)
 - AfskrevetFordring
 - **FordringUnikIdentifikation**
- FordringAfskrivningNedreguleringListe (0 eller 1)
 - FordringAfskrivningNedregulering(0 eller flere)
 - AfskrevetFordring
 - **FordringUnikIdentifikation**
 - NedreguleretFordringAfskrivning
 - **FordringAfskrivningUnikIdentifikation**
- FordringDækningListe (0 eller 1)
 - DækketFordring(0 eller flere)
 - **FordringUnikIdentifikation**
- FordringDækningNedreguleringListe (0 eller 1)
 - FordringDækningNedregulering(0 eller flere)
 - DækketFordring
 - **FordringUnikIdentifikation**
 - NedreguleretFordringDækning
 - **FordringDækningUnikIdentifikation**

Du skal således implementere kode, der aflæser disse og gemmer i en midlertidig transaktionstabel eller lignende. Således at disse er låst for opdatering, så længe forespørgsel processeres. Inden videre processering skal du tjekke, om der allerede findes en igangværende transaktion indeholdende samme primære objekter. Og afvise forespørgsel, hvis dette er tilfældet. Her fra listen af Fejlårsagskoder i servicebeskrivelsen:

UUID	Brugervendt Nøgle	Titel	Beskrivelse
bf6ca1c4-14a6-4f87-ada8-5c28ce26db6f	01.0003.016	En anden dataleverance med samme primære informationsobjekt er under databehandling i servicen.	En anden dataleverance med samme primære informationsobjekt er under databehandling i servicen.

Kontrol af genfremsendelse

Hvis der findes en tidligere *afsluttet* transaktion med samme primære objekter, venligst se servicebeskrivelsen for detaljer vedrørende dette. De relevante Fejlårsagskoder er:

a9157c39-79d9-4aec-9a9f-97edd4ae5db3	01.0003.014	Transaktionen er tidligere modtaget. Der findes et forretnings svar, som ikke kan fremsendes pga. strukturændring af servicen.	Transaktionen er tidligere modtaget. Der findes et forretnings svar, som ikke kan fremsendes pga. strukturændring af servicen.
61c69e0d-5f31-409e-a6c3-866a5b650923	01.0003.015	Transaktionen er tidligere modtaget. Status for dataoverdragelsen er ukendt.	Transaktionen er tidligere modtaget. Status for dataoverdragelsen er ukendt.

5.6.3. Forretningsmæssig validering

Når den generelle leverancekontrol er gennemført med succes, da skal du udføre den forretningsmæssige validering. Igen her, venligst se servicebeskrivelsen for detaljer. Her et udsnit af regler:

1. Debitorkonto
 - 1.1. Er registreret i fagsystemet (II)
 - 1.2. Systemet har tilladelse til at udføre operationen på kontoen (III)

4. FordringDækning
 - 4.1. Dækning må ikke tidligere være registreret (FordringDækningUnikIdentifikation) (II)
 - 4.2. OprettetDatoTid må ikke ligge frem i tid (II)
 - 4.3. Fordring - FordringUnikIdentifikation
 - 4.3.1. Skal være registreret tidligere (II)
 - 4.3.2. Er angivet under debitor kontoen (jf. 1) (II)

Og her eksempler på tilhørende Fejlårsagskoder:

b70f64fc-4882-472c-90ad-c68b49a0ac61	02.0005.001	Debitorkonto findes ikke	Debitorkonto findes ikke
3b9101dd-a7aa-42b5-b815-2ea9c853ec14	02.0005.002	Systemet har ikke ret til at udføre den ønskede handling på debitor kontoen	Systemet har ikke ret til at udføre den ønskede handling på debitor kontoen

Vi har nu implementeret validering, og er klar til teste.

5.6.4. Eksempel - dækning

Her et fabrikeret eksempel hvor debtorsystemet orienterer om indbetaling (dækning) på en fordring. Husk, at den rå forespørgsel er indpakket i SOAP <Envelope><Body>.

Vi starter med at implementere og teste modtagelse af dækning på en fordring. Vi laver en testbesked indeholdende elementer markeret med fed:

- **OpretDækningAfskrivning_I**
 - **HovedOplysninger**
 - **DebitorkontoUnikIdentifikation**
 - FordringAfskrivningListe
 - FordringAfskrivningNedreguleringListe
 - **FordringDækningListe**
 - **FordringDækning**
 - FordringDækningNedreguleringListe

Vi starter med en simpel dækning af det fulde beløb på den fordring vi oprettede tidligere. Vi sender en valid forespørgsel og tjekker, at det virker og dækning registreres korrekt.

OpretDækningAfskrivning_I eksempel

(SOAP-elementer udeladt)

```
<OpretDækningAfskrivning_I>
  <HovedOplysninger>
    <TransaktionsId>9ad2e646-2a9e-4332-86b8-18ccaf153aff</TransaktionsId>
    <TransaktionsTid>2022-09-19T12:00:23.922+02:00</TransaktionsTid>
    <Rute>
      <AfsenderOrganisation>11111111</AfsenderOrganisation>
      <AfsenderItSystemInstans>062b5b53-1dcb-40d8-8580-
b309f487fb02</AfsenderItSystemInstans>
      <ModtagerOrganisation>11111111</ModtagerOrganisation>
      <ModtagerItSystemInstans>f269ca0c-7b5d-49e2-a585-
1ee39ba5a431</ModtagerItSystemInstans>
    </Rute>
  </HovedOplysninger>
  <DebitorkontoUnikIdentifikation>41d54894-a515-4b75-8754-
bbab3d756ff7</DebitorkontoUnikIdentifikation>
  <ns:FordringDækningListe>
    <ns:FordringDækning>
      <ns:FordringDækningUnikIdentifikation>f69c0c4d-cc47-4d6e-88d4-
f158f6d50fbf</ns:FordringDækningUnikIdentifikation>
      <ns:FordringDækningOprettetDatoTid>2022-09-
18T11:00:23.922+02:00</ns:FordringDækningOprettetDatoTid>
      <ns:FordringDækningBeløb>2000</ns:FordringDækningBeløb>
      <ns:Indbetaling>
        <ns:IndbetalingBetalingsdato>2022-09-18</ns:IndbetalingBetalingsdato>
      </ns:Indbetaling>
      <ns:DækketFordring>
        <ns:FordringUnikIdentifikation>7cd86df8-306b-407e-ae16-
7a3901554e5d</ns:FordringUnikIdentifikation>
      </ns:DækketFordring>
    </ns:FordringDækning>
  </ns:FordringDækningListe>
</OpretDækningAfskrivning_I>
```

- Vi aflæser TransaktionsId og TransaktionsTid til logning og til returnering i svaret.
- AfsenderItSystemInstans er her UUID for debtorsystemet.
- DebitorKontoUniklIdentifikation er den vi oprettede tidligere.
- FordringDækningUniklIdentifikation defineres af debtorsystemet
- FordringDækningBeløb her samme beløb som vi registrerede fordringen på. Du bør også teste, fx at modtage to dækninger på hver 1000 kr.
- FordringUniklIdentifikation er den vi oprettede tidligere.

Når dette eksempel virker, fortsæt da med at teste de andre opdateringer:

- FordringDækningNedreguleringListe
- FordringAfskrivningListe
- FordringAfskrivningNedreguleringListe

5.6.5. Test validering

Opret test cases for hver valideringsregel og udfør dem. Eksempel: Lav en ændring til eksempel i forrige afsnit, således at XML ikke er valid, og send til service. Tjek at dit system returnerer "01.0001.001 Leverancen kan ikke skemavalideres".

Du skal også lave test cases der tjekker, at din service korrekt håndterer flere brud på forretningsregler i samme forespørgsel, samt at din service er idempotent. Fx i tre trin:

1. Send en forespørgsel med dækning på to fordringer med fejl i begge - fx . Tjek at begge fejl korrekt returneres.
2. Send ny forespørgsel hvor fejl i første dækning er rettet. Tjek at første dækning registreres korrekt, mens at anden dækning ignoreres og stadig rapporterer fejlbehæftet.
3. Send ny forespørgsel med begge fejl rettet. Tjek at første dækning ignoreres med advarsel om at den allerede er modtaget tidligere (*se genfremsendelse fejlårsagskode i forrige afsnit*), samt at anden dækning nu korrekt registreres.
4. Send eventuelt den korrekte forespørgsel igen, og tjek at begge ignoreres med genfremsendelse returfejl.

Sidst, men ikke mindst, skal vi også teste, at vi korrekt kan håndtere "Sikring mod parallel håndtering af samme primære informationsobjekt". Eventuelt indføre en timer i koden (efter den generelle leverancekontrol) i testøjemed, der venter, således at processeringen tager tilstrækkeligt tid, til at dette kan testes. Mens første forespørgsel kører sender du samme forespørgsel afsted fra en anden klient, og tjekker at denne bliver afvist.

VERSIONSHISTORIK

Version	Dato	Ændringer
1.0	05-10-2022	Første version
2.0	18-09-2024	Tilpasset til version Version 4 af integrationen SF1590_B_01
2.1	24-03-2025	Små justeringer
2.2	19-12-2025	Små justeringer