



capevo
business process evolution

www.capevo.com

Webservice

Indholdsfortegnelse

Indledning.....	3
Webservicetyper.....	4
Opsætning af webservice	6
Særlige nøgleord.....	8
EBML til udgående XML.....	11
Svar fra webservices	12
Onload og Interactive	13
OnShow	14
Push.....	14
DataList.....	15
FieldValidation.....	16
Admin Logon	18
Fejlhåndtering.....	19
Tilknyt webservice til formular.....	20
Mapping	20

Indledning

XML-baserede webservices er kernen i Capevo XForms udveksling af oplysninger med andre systemer – uanset om det sker i forbindelse med udfyldelse af en formular eller som led i en dataoverførsel.

Denne vejledning forudsætter kendskab til webservices på ekspertniveau.

Webservicetyper

I Capevo XForm er webservices opdelt i forskellige kategorier, ud fra hvilken funktion de varetager:

OnLoad

Ved skift mellem formularsider forudfyldes et eller flere felter med en værdi der modtages fra en webservice.

Interactive

Er af natur ligesom en OnLoad, dog køres denne ikke ved skift af formularside, men når brugeren forlader et formularfelt. Ud fra det pågældende formularfelts værdi, kan webservicen returnere data til forudfyldelse i andre felter på formularsiden.

FieldValidation

Modtager værdier fra udvalgte felter og kontrollerer om værdierne passer med et bestemt kriterie angivet i webservicen. Hvis ikke, kan FieldValidation foretage sig en eller flere af følgende:

1. Returnere en fejlbesked
2. Returnere en URL som brugeren skal redigeres til.
3. Returnere feltnavne som skal fremhæves med gult.
4. Returnere en liste over valgmuligheder på et bestemt felt, som bliver repræsenteret af en rullemenu.

DataList

Køres samtidig med OnLoad, ved skift af formularside. Webservicen udfylder en liste over svarmuligheder i en rullemenu eller rulleliste.

OnShow

Afgør om en bestemt formularside skal vises. Hvis formularsiden ikke skal vises, springes den over. Der kan ikke være en OnShow på sidste side.

Push

Når formularen indsendes kan der sendes udvalgte informationer til en webservice. Alternativt kan hele formularens indhold medsendes. Push kan bl.a. returnere en statuskode, en kvitteringstekst og et modtagested.

AdminLogon

Benyttes når sagsbehandlere skal foretage brugersimulering på formularer som kræver logon med den fælles pinkode (KSP/WEB). Webservicen modtager brugernavn og adgangskode som sagsbehandleren indtaster ved brugersimulering.

Standard webservices

I Capevo XForm er der oprettet en række standard webservices, som mange vil kunne bruge, f.eks. postnummeropslag, CVR- og CPR-opslag m.m. Hvis du ønsker at benytte standard-webservices og ikke kender indholdet af de enkelte webservices, kan du kontakte os og blive oplyst om detaljerne.

Opsætning af webservice

En webservice opsættes af en administrator under fanebladet "Webservices". De webservices som opsættes her, kan tilknyttes alle formularer.

Til opsætning af en webservice, som er udviklet med .NET, kan man med fordel besøge asmx-filen for webservicen, klikke på metodens navn og se de oplysninger man skal benytte for at kunne opsætte webservicen.

Service er aktiv

Hvis dette felt ikke er afkrydset, vil webservicen ikke blive eksekveret, selvom den er tilknyttet en formular. Når man tilknytter webservice viser et grønt og et rødt ikon om webservicen er henholdsvis aktiv eller ikke aktiv.

Service type

Her vælges den type webservice som der opsættes. Dette har betydning for tidspunktet hvor webservicen eksekveres og hvordan svaret bliver fortolket.

Service navn

Et sigende navn som du selv kan bruge til at genkende webservicen. Dette navn bliver i nogle tilfælde vist for personen der udfylder formularen, i det tilfælde at din webservice går ned.

EndPointUrl

Den url-adresse som der skal POST'es til. Hvis det er en .NET-webservice du er ved at sætte op, skal du her skrive adressen til asmx-filen.

Action

Den SOAP Action på webservicen der skal udføres. Det der står efter den sidste skråstreg i din SOAP Action, er en del af det svar som webservicen forventer at få tilbage.

Max antal kald i timen pr. session

Hvis du vil forhindre i at din webservice bliver misbrugt, f.eks. hvis den belaster systemet eller koster penge for hvert fald, kan du her skrive hvor mange gange en givet bruger må benytte webservicen pr. session. Det vil sige, dette tal nulstilles for hver bruger, hver gang personen åbner et nyt browservindue.

ErrorNode

Her kan du skrive navnet på en node som systemet skal kigge efter, for at identificere om der er opstået en fejl. Default er "ErrorMessage", hvilket vil betyde at systemet kigger efter om der er indhold i en XML-node der ser sådan ud:

```
<ErrorMessage>Dette er en fejl</ErrorMessage>
```

Send email ved fejl til

Indtast en e-mail-adresse som skal modtage eventuelle fejlrapporter. I fejlrapporten er der typisk det sendte request og et response hvis et sådant eksisterer.

Fortsæt ved fejl

Hvis dette felt er afkrydset, bliver brugeren ikke bremset af at der opstår fejl i en webservice. Dette kan være relevant, hvis webservicen kun assisterer brugeren i udfyldelse af formularen, men ikke er direkte nødvendig for indsendelsen.

Brugeren modtager stadig fejlmeddelelser selvom *Fortsæt ved fejl* er afkrydset. Ved Push ser brugeren ikke fejlmeddelelsen – den skrives i stedet i den interne note for den indsendte formular.

Trigger field

Hvis webservicen er af typen "Interactive", skal man indtaste et Trigger field. Her indtastes *Felt ID* på det felt som skal aktivere webservicen. Når dette felt forlades, på den formular hvor webservicen tilknyttedes, bliver webservicen kaldt.

Push ved indsendelse

Hvis webservicen er af typen "Push", kan man vælge om webservicen skal udføres når formularen indsendes. I modsat fald, bliver webservicen kun kaldt hvis en XForm-bruger åbner den indsendte formular og klikker på ikonet oppe i højre hjørne, med teksten "Overfør til fagsystem".

Timeouts i sekunder

Her er det muligt at angive hvor lang tid der må gå før formularen opgiver at komme i kontakt med webservicen. Det kan være du ikke er i tvivl om, at hvis der er gået 5 sekunder, så er der et eller andet galt, og så kan du skåne brugeren for yderligere ventetid ved at sætte et timeout på 5 sekunder på.

Envelope

Her skrives den tekst som skal sendes til webservicen. Feltet hedder Envelope, da indholdet typisk er en SOAP Envelope (XML).

I dette felt er det muligt at skrive nøgleord, i dette format:

```
<node>[#nøgleord#]</node>
```

Et nøgleord kan være et *Felt ID*, hvilket vil betyde at indholdet af det givne felt, vil blive indsat i din Envelope. Desuden kan du benytte særlige nøgleord.

Nøgleord til Interactive

Hvis der benyttes en webservice af typen Interactive, er det indhold der kan sendes til webservicen, indholdet af det felt som er skrevet under "Trigger field". For at definere hvor i din Envelope, dette indhold skal indsættes, skal du benytte følgende nøgleord:

```
[#TRIGGER_FIELD_VALUE#]
```

Særlige nøgleord

Ud over formularernes feltnavne kan der benyttes en række særlige nøgleord i definitionen af webservicekald. En del af disse nøgleord fungerer kun i særlige situationer. F.eks. vil det ikke give mening at benytte *PaymentTransactionNumber* hvis der ikke er betaling på den pågældende formular.

- KMD_LOGON_CPR: F.eks. P1111111111
- KMD_LOGON_TYPE: None, Unknown, Pin, Certificate, Simulated, Edit
- FormHTML: Al html i hele formularen. Det lægges ind i en <![CDATA[]] blok.
- FormHTMLBase64: Al html i hele formularen som base64
- FormXML: Alle formularens værdier sat ind som XML.

- FormXMLBase64: Formularens værdier i XML og base64-encoded.
- FormID: Et unikt database id der identificerer formulartypen.
- Attachments: Inkluderer XML som indkapsler vedhæftede filer.
- AttachmentsAsString: Inkluderer en streng som indeholder filnavne og selve de vedhæftede filer som Base64.
- SessionID: Id fra brugerens session.
- FormTitle: Titlen på formulartypen.

- FormAlias: Alias på formulartypen.
- Advisering: Hvorvidt der skal adviseres om indsendelsen.
- ReferenceNumber: Et referencenummer der er unikt for hver indsendelse.
- PaymentCurrency: Den valuta betalingen foregår i.
- PaymentAmount: Beløbet som er betalt eller forventes betalt.
- PaymentTransactionNumber: Transaktionsnummeret fra DIBS efter en betaling.
- PaymentDatetime: Tidspunktet for en betaling.
- PaymentIsPaid: Hvorvidt der er betalt.
- PaymentName: Navnet på den valgte betaling.
- PaymentType: Typenavnet på den valgte betaling.
- ReceiptEmail: Kvitterings-emailen valgt af brugerne.
- IsContentSensitive: : true eller false, alt afhængigt af om formularen indeholder personfølsomme oplysninger.
- LogonUsername: Brugernavn der benyttes til at tilgå formularen (af sagsbehandler).
- LogonPassword: Adgangskode der benyttes til at tilgå formularen (af sagsbehandler).
- SubmissionYear: Året for indsendelse af formularen. F.eks. 2005
- SubmissionMonth: Måned for indsendelse af formularen. F.eks. 06
- SubmissionDay: Dagen for indsendelse af formularen. F.eks. 15
- ClientCertSubject: Hele subject uredigeret fra certifikatet der blev brugt til digital signatur.
- ClientCertSubjectPID: PID fra certifikatet, til at identificere brugeren.
- SurveyRecipientGroup: Den respondentgruppe som brugeren tilhører.
- SurveyRecipientGroupShort: Det korte respondentgruppenavn.
- PrintPDFFormHtml: Returnerer en PDF
- KMD_LOGON_CPR
- KMD_LOGON_TYPE
- FormHTML
- FormHTMLBase64
- FormXML
- FormXMLBase64
- FormXMLEncoded
- FormID: Returnerer formularens ID
- Attachments
- AttachmentsAsString
- AttachmentNames
- SessionID

- FormTitle: Returnerer formularens titel
- FormAlias: Returnerer formularens alias
- Advisering
- ReferenceNummer
- ReferenceNumber
- PaymentCurrency: Returnerer betalingsvalutaen
- PaymentAmount: Returnerer betalingsbeløbet
- PaymentTransactionNumber: Returnerer betalingens transaktionsnummer
- PaymentDatetime: Returnerer datoen for betalingen
- PaymentIsPaid
- PaymentName
- PaymentType
- ReceiptEmail: returnerer kvitteringsemailen
- IsContentSensitive
- LogonUsername
- LogonPassword
- SimulatingUserUsername
- SimulatingUserPassword
- CaseFolderUserGroup
- CaseFolderUserType
- CaseFolderUserGroupDisplayName
- SubmissionYear
- SubmissionMonth
- SubmissionDay
- StatusCode
- StatusCodeText
- CertificateBase64
- CertificateSubject
- ClientCertSubject
- ClientCertSubjectPID
- SurveyRecipientGroup
- SurveyRecipientGroupShort
- SkippedPageNumbers
- ValidationResult
- MetaData
- PrintPDFFormHtml

EBML til udgående XML

Det er muligt at ændre værdien af et felt i sidste øjeblik, ved at benytte en særlig syntaks i den XML man opsætter i webservicen.

F.eks. giver en checkboks resultatet "on" hvis den er afkrydset og resultatet "" hvis den ikke er afkrydset. Hvis man i sin webservice hellere vil modtage "true" og "false" eller "0" og "1", kan man ændre værdien inden den sendes til webservicen.

Eksempler

Normalt hentes værdien af checkboksen ud sådan:

```
[#min_checkboks#]
```

Men med EBML kan man ændre værdien sådan:

```
[#min_checkboks#on:true;:false]
```

Hvis du f.eks. har en rullemenu som enten kan give værdierne "Ja" eller "Nej", men hellere vil modtage "+" og "-" i din webservice, kan du skrive følgende:

```
[#min_rullemenu#Ja:+;Nej:-]
```

Svar fra webservices

I dette afsnit er det defineret hvilket format der forventes ved et web service svar fra eksterne systemer.

Alle informationer skal lægges i en node der hedder MetodeResponse, hvor Metode er navnet på den givne Metode defineret efter sidste '/' i SOAPAction, hvor webservicen er opsat.

Hvis SOAPAction ser sådan ud: "http://soap.dwb.dk/ws/MinMetode", er metodens navn: MinMetode. Bemærk i øvrigt at 'Action' ved opsætning af en .NET-webservice er det der står ud for SOAPAction.

Eksempel på gyldigt svar, hvor Metoden hedder "MinMetode":

```
POST /ws/MineWebServices.asmx HTTP/1.1
Host: 111.111.111.111
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://soap.dwb.dk/ws/MinMetode"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <MinMetodeResponse xmlns="http://www.ditnamespace.dk/ws/">
      <MinMetodeResponseResult>string</MinMetodeResponseResult>
    </MinMetodeResponse>
  </soap:Body>
</soap:Envelope>
```

Det, der er interessant, er det der står indenfor <MinMetodeResponse></MinMetodeResponse>. Alt det udenom (selve SOAP-kuverten) er principielt uden betydning for integrationen, blot det overholder XML-standarden.

Onload og Interactive

Disse webservice-typer kan returnere XML-noder med indhold, som bliver udfyldt i formularfelter.

Eksempel på et gyldigt svar:

```
<MinMetodeResponse xmlns="http://www.ditnamespace.dk/ws/">
  <Person>
    <Name>Anders</Name>
    <Age>20</Age>
  </Person>
</MinMetodeResponse>
```

Formulardesigneren har indsat to felter på formularen: Navn og Alder

Formulardesigneren sørger for at mappe *Name* til *Navn* og *Age* til *Alder*. I det givne eksempel vil der stå "Anders" i Navn-feltet og "20" i Alder-feltet.

Bemærk at noderne befinder sig i en node der hedder "MinMetodeResponse". Dette er under antagelsen af SOAP Action er "MinMetode". Årsagen til at dette format forventes er, at det er den måde .NET-webservices fungerer på som standard.

Formularen kigger efter de felter som indeholder tekst. Det har ingen betydning i ovenstående eksempel, at de to XML-noder befinder sig i noden <Person>. Strukturen under <MinMetodeResponse> er underordnet, så længe der er tale om gyldig XML.

Særlige noder ved OnLoad

Følgende to noder kan som noget særligt benyttes i en OnLoad:

<RECEIPT_MAIL> forudfylder e-mail-adressen som angiver hvor kvitteringen skal sendes til.

<PAYMENT_AMOUNT> angiver det beløb som skal betales, ved en evt. betaling.

OnShow

Føgende svar forventes fra en OnShow-webservice.

```
<MinMetodeResponse xmlns="http://www.ditnamespace.dk/ws/">
  <MinMetodeResult>boolean</MinMetodeResult>
</MinMetodeResponse>
```

Der kigges efter indholdet af <MinMetodeResult>, som skal være enten *true* eller *false*. Værdien angiver om den givne side som webservicen er tilknyttet, skal vises eller ej. Hvis boolean er true, bliver den næste side vist, ellers bliver den ikke vist.

Push

Ved indsendelse af formularen kan der pushes data til en webservice. Svaret på webservicen kan påvirke følgende:

1. En statuskode som skal gemmes på formularen.
2. Den kvitteringstekst der skal vises for brugeren.
3. En identifikator på det sted der skal modtage den indsendte formular.
4. En intern note med eventuelle kommentarer til den indsendte formular. Her skrives eventuelle fejlmeddelelser også.

Et gyldigt svar ser sådan ud:

```
<MinMetodeResponse xmlns="http://www.ditnamespace.dk/ws/">
  <MinMetodeResult>
    <Status>int</Status>
    <KvitteringScreen>string</KvitteringScreen>
    <KvitteringMail>string</KvitteringMail>
    <Kvittering>string</Kvittering>
    <Modtagested>string</Modtagested>
    <Note>string</Note>
  </MinMetodeResult>
</MinMetodeResponse>
```

Hvis <Kvittering> sendes med, gælder den for både <KvitteringScreen> og <KvitteringMail>

DataList

Her er et eksempel på et gyldigt svar fra en webservice, som skal angive de valgmuligheder brugeren skal have i en rullemenu.

```
<MinMetodeResponse xmlns="http://www.ditnamespace.dk/ws/">
  <MinMetodeResult>
    <Option>
      <Value>string</Value>
      <Text>string</Text>
    </Option>
    <Option>
      <Value>string</Value>
      <Text>string</Text>
    </Option>
  </MinMetodeResult>
</MinMetodeResponse>
```

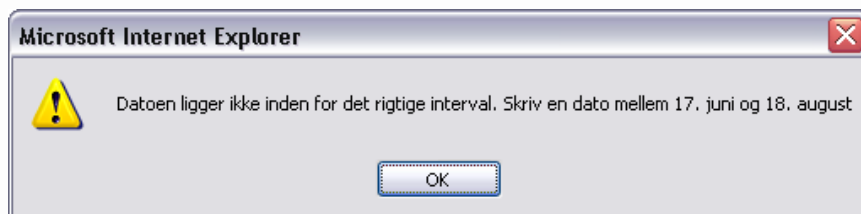
Value under Option er den værdi datalisten har, hvis valgmuligheden er valgt. Text er den valgmulighed der vises for brugeren.

Der skal være en <Option>-node for hver valgmulighed der skal være i rullemenuen.

FieldValidation

Denne udføres når der bladres væk fra en side. Formularen sender typisk noget feltindhold til webservicen og webservicen validerer feltindholdet efter noget avanceret logik i webservicen. Hvis feltindholdet ikke er gyldigt, har webservicen forskellige muligheder for at gøre brugeren opmærksom på det.

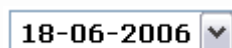
- Skriv en *Alert* til brugeren med en beskrivelse af hvad der er galt og hvad der skal gøres for at rette det.



- Fremhæv et felt med gul baggrund.



- Udskift feltet med en rullemenu.



Advarslen vises på samme måde som ved andre webservices, ved hjælp af et `<ErrorMessage>`-tag. Ved andre webservices bliver resten af svaret ignoreret, hvis der er angivet en `ErrorMessage`, men det er ikke tilfældet ved `FieldValidation`. Her kan man godt returnere en fejlmeddelelse, og stadig benytte nogle af webservicens andre funktioner, f.eks. farve et felt gult.

For at farve et felt gult, skal man angive hvilket felt der skal farves gult. Det gøres sådan:

```
<Inputs>
  <Input>
    <Name>feltnavn</Name>
  </Input>
  <Input>
    <Name>feltnavn</Name>
  </Input>
</Inputs>
```

Hvis man ønsker at give valgmuligheder for brugeren i form af en rullemenu, skal man under hvert <Input> angive en række <Option>'s, sådan:

```
<Inputs>
  <Input>
    <Name>feltnavn</Name>
    <Options>
      <Option>
        <Value>feltværdi</Value>
        <Text>felttekst</Text>
      </Option>
      <Option>
        <Value>feltværdi</Value>
        <Text>felttekst</Text>
      </Option>
    </Options>
  </Input>
</Inputs>
```

Der kan mappes mellem indholdet af <Name></Name> og formularens felter.

Admin Logon

Denne benyttes til fra administrationsmodulet at give adgang til brugersimulering og "Overførsel til fagsystemer". Det der sendes til webservicen vil typisk bestå af et brugernavn og en adgangskode. Der benyttes [#LogonUsername#] og [#LogonPassword#] som nøgleord i XML'en.

Her er det svar der forventes fra webservicen:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <ErrorHandler xmlns="http://tempuri.org/">
      <ErrorMessage>string</ErrorMessage>
    </ErrorHandler>
  </soap:Header>
  <soap:Body>
    <MinMetodeResponse xmlns="http://tempuri.org/">
      <MinMetodeResult>boolean</MinMetodeResult>
    </MinMetodeResponse>
  </soap:Body>
</soap:Envelope>
```

Hvis *boolean* er true, betyder det at brugernavn og adgangskode er godkendt og der gives adgang til funktionen.

ErrorMessage behøver ikke være med. Hvis ErrorMessage er med, bliver resten ignoreret. Det betragtes som et mislykket logon. Hvis der er en ErrorUrl med, bliver den ignoreret.

Fejlhåndtering

Ved alle typer webservices, er det muligt for webservicen at returnere en bestemt fejl-tekst, som skal benyttes når webservicen fejler. Det er desuden muligt at angive en URL som brugeren skal overføres til.

```
<ErrorMessage>Der er sket en fejl</ErrorMessage>  
<ErrorUrl>http://www.google.com</ErrorUrl>
```

Hvis der returneres en <ErrorMessage> fra en webservice, bliver resten af webservicens svar ignoreret (Med undtagelse af FieldValidation webservice).

Indholdet af ErrorMessage bliver vist for brugeren i en javascript alert. Hvis "Fortsæt ved fejl" er slået til på webservicen og der er tale om en Push-webservice, bliver indholdet af ErrorMessage i stedet skrevet i den interne note for den pågældende indsendte formular.

Tilknyt webservice til formular

Hvis det skal være muligt at tilknytte en webservice til en formular, skal webservicen på forhånd være opsat under *Admin*. Webservices tilknyttes under *Indstillinger* for en formular. Derunder er et faneblad der hedder *Webservices*.

Der kan du vælge mellem alle webservices som er opsat, i ruden til venstre. Ved at trykke på Plus-ikonet ud for en webservice, bliver den tilknyttet din formular.

En Push-webservice tilknyttes selve formularen, mens de øvrige webservice tilknyttes bestemte formular-sider.

Mapping

Ved Onload, Interactive, FieldValidation og DataList, skal der mappes mellem webservicens felter og formularens felter. Det gøres ved at klikke på XML-ikonet ud for webservicen i den højre rude.

Derunder kan du vælge mellem de formularfelter der er på siden. Desuden kan du indtaste et navn på en XML-node. Du skal indtaste navnet på den XML-node hvis indhold skal overføres til det felt som du har valgt.

Ved FieldValidation skal du indtaste det som står i <Name>-noden.

Ved DataList skal du indtaste *MinMetodeResult*, afhængigt af hvad SOAP Action er på den givne webservice.