

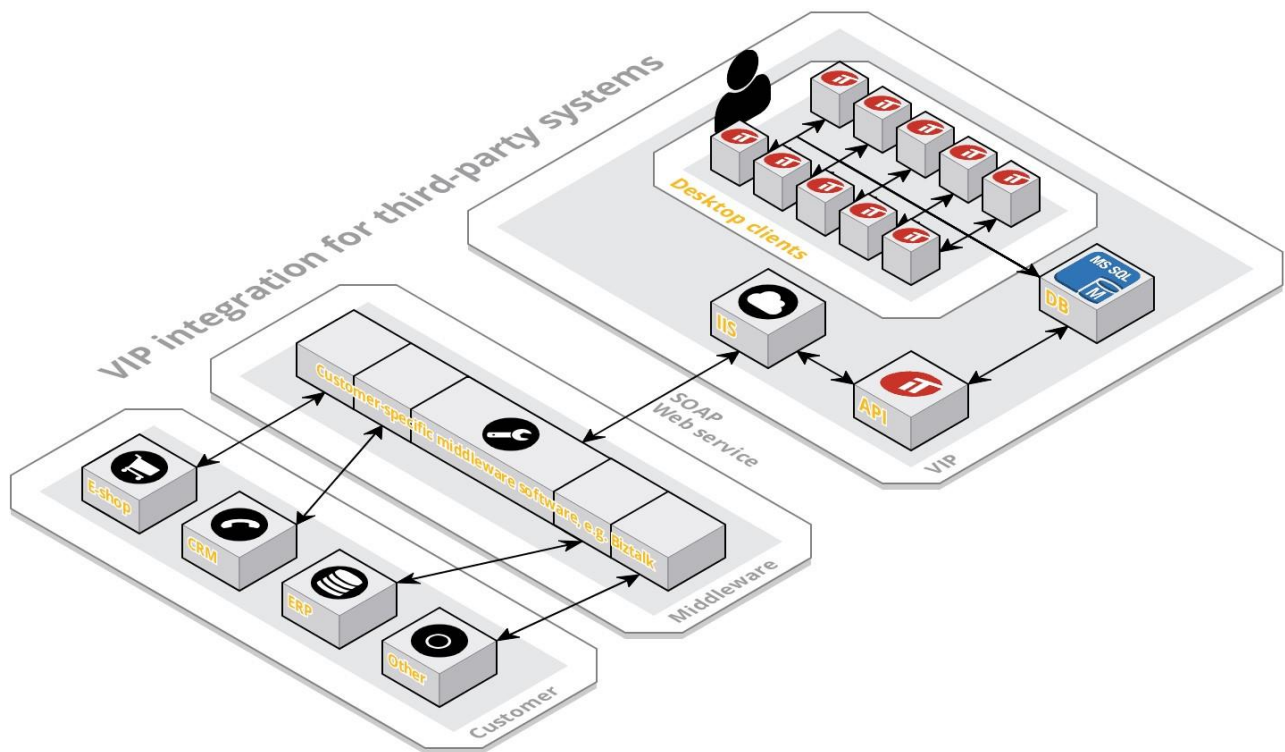
Dokumentation til Standard API version 2

Indhold

Teknisk oversigt	2
Rowversion	4
Xsi:nil	5
Eksempel på et flow	6
Response	6
CreateWebOrder/Quote	8
CalcOrder/Quote	14
GetOrder/Quote	16
DeleteOrder/Quote	17
GetRevisedOrderList/QuoteList	19
CalcModel	20
GetOrderPrint/QuotePrint	20
GetOrderPositionPrintImage	21
UpDateOrderHead	21
WorkFlowInput	22
Get/SetUserDefinedFields	22
MasterDataMaintenance	27
GetProductionData	29
C#	30

Teknisk oversigt

Ved hjælp af Standard API funktionerne i VIP, er det muligt at kommunikere med andre systemer, f.eks. ERP, CRM, Websalg osv.

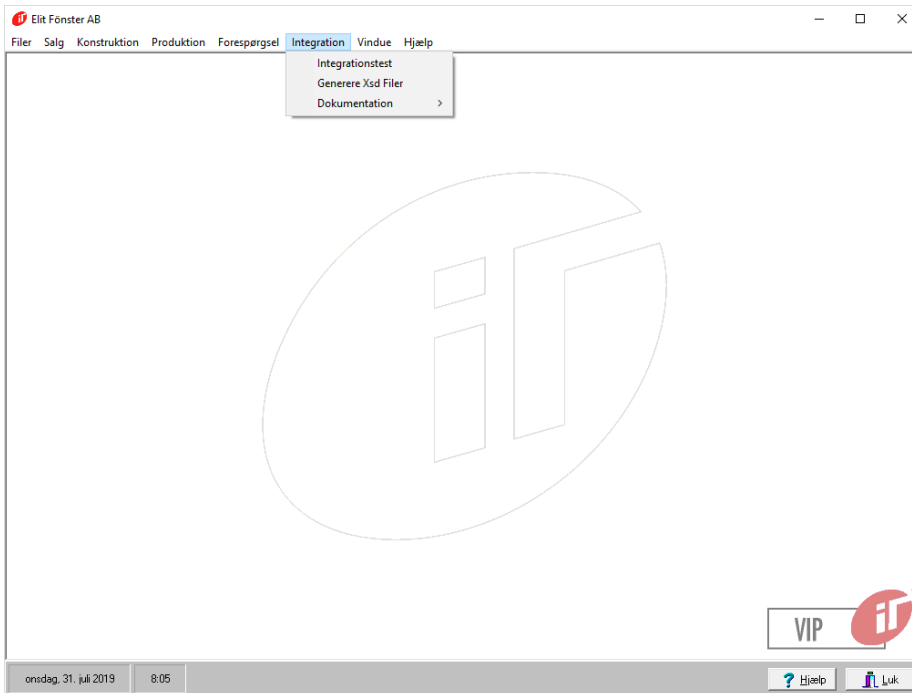


Microsoft IIS anvendes til driften af VIP's webservice, som består af en SOAP API funktion (request).

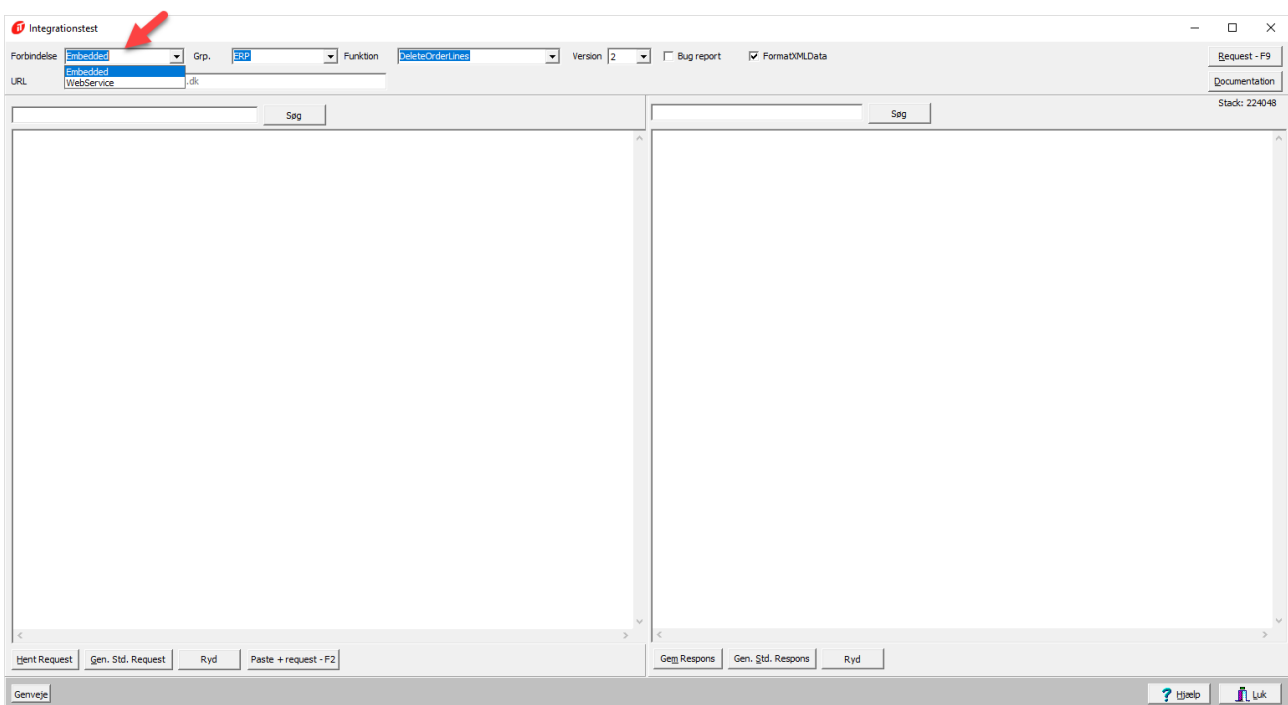
Request-funktionen modtager 2 parametre:

- Funktionsnavn (Tekststreng) som angiver hvilken VIP API funktion der skal anvendes til dette request.
- Request-data (XML) som beskriver dataene der skal behandles.

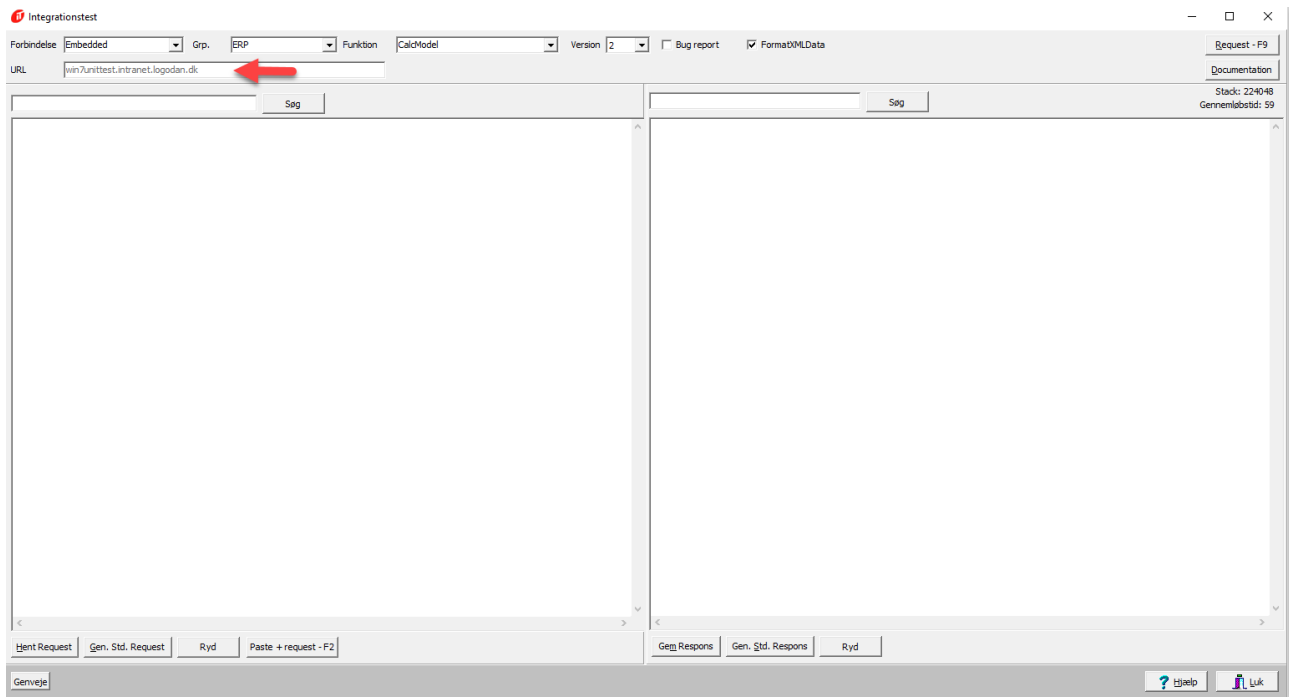
Der findes XSD skema og en teknisk dokumentation til alle Standard API funktioner. XSD'erne og dokumentationen kan hentes fra VIP under menupunktet **Integration**. I den tekniske dokumentation er alle taggene beskrevet med feltnavnet i VIP, felttype og længde.



I menuen findes der en Integrationstestform, hvor det er muligt at teste funktionerne både som Embedded (kald direkte til VIP) og som Webservice, hvor kaldet går gennem IIS serveren.



Ønskes der at testes gennem webservicen, skal hostnavnet skrives ind i feltet **URL**.



Det anbefales at udvikle op mod de funktioner der ligger under Version2, da Version1 ikke bliver udvidet med nye funktioner, og vil på sigt udgå.

Alle funktioner der findes til tilbud findes også til ordre, og funktionerne fungerer og indeholder de samme informationer uanset om det er tilbud eller ordre de udvikles til.

Der vil ikke blive ændret i betydningen af tags eller fjernet tags, ligesom der heller ikke vil blive ændret i strukturen ved at tilføje ekstra lag. Der vil dog kunne blive tilføjet nye tags til eksisterende funktioner.

Alle kald logger request og response i mapper under log mappen. Der vil automatisk blive oprettet en mappe når en funktion tages i brug. Mappen vil være navngivet Integ_XXXX (navnet på den gruppe funktionen tilhører).

Rowversion

Mange af funktionerne i Standard API anvender rowversion, som er udviklet af Microsoft. For yderligere oplysninger om Rowversion, klik [her](#).

Rowversion er et unikt nummer der automatisk bliver påsat en tabelrække, når der er foretaget en ændring af data. Dette anvendes i Standard API'et til bla. at returnere en liste over nyligt ændrede ordrehoveder, positioner osv. Der bliver ikke genereret en ny række i databasen hver gang rowversion ændres, der findes

kun én række, så rowversion bliver over skrevet. Dette betyder, at der ikke kan hentes tidligere versioner af en række, det er altid den sidst nye række der hentes.

Alle response fra Standard API'et returnerer et Rowversion nummer i tagget `<RowVersion>`.

Eksempel på anvendelse af rowversion:

GetRevisedOrderList (Returnerer en liste over alle ordre der er foretaget ændringer på):

I requesten angives Rowversion:

```
<?xml version="1.0" encoding="UTF-16"?>
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <FromRowVersion>4696251</FromRowVersion>
  <MaxRecordCount/>
  <IncludeDeleted/>
</Request>
```

Hvis der ikke er angivet andet i requesten vil der blive returneret en liste over ordre der har et højere Rowversion end det er der angivet.

Det højeste Rowversionnummer +1 der bliver returneret skal så anvendes til at kalde funktionen med næste gang.

Xsi:nil

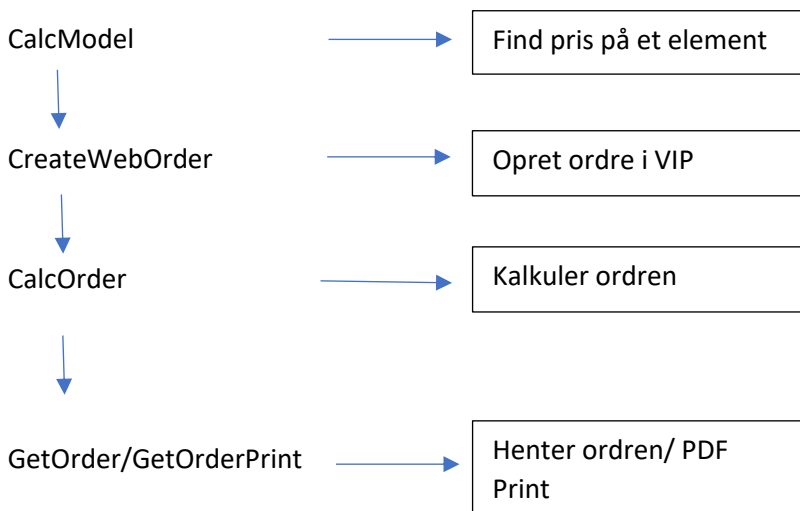
Xsi:nil attributten indikerer om et element har en værdi eller om værdien er ukendt. Der er forskel på værdien 0 og ingen værdi.

Hvis et tag er skrevet som `<Tagname xsi:nil="true"/>` ændrer det ikke på værdien i feltet, svarende til at man helt undlader tagget.

Hvis et tag er skrevet som `<Tagname xsi:nil="false"/>` blanker det værdien i feltet, svarende til at man helt skriver et tomt tag. `<Tagname/>` eller `<Tagname></Tagname>`

Eksempel på et flow

Eksemplet viser hvordan et muligt flow til et website kan integreres.



Response

Alle response er inkluderet error- response kode samt generel information vedrørende systemet, bla. hvilken version der anvendes og forbrugt tid på eksekvering af requesten.

```

<Response xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ErrorCode>OK</ErrorCode>
  <ErrorName>OK</ErrorName>
  <ErrorDescription/>
  <ResponseCode>OK</ResponseCode>
  <ResponseDescription>Model is calculated</ResponseDescription>
  <ResponseCodeGroup>OK</ResponseCodeGroup>
  <ResponseInfo/>
  <SystemInfo>
    <TSystemInfo>
      <Version>17.70.0.T</Version>
      <Revision>25066</Revision>
      <CompileDate>03-09-2019 13:45:19</CompileDate>
      <WindowsUserName>lcg</WindowsUserName>
      <Module>Vip.exe</Module>
      <Database>VIP</Database>
      <SetupDrvPath>\\lgudrvfil\VIPPROG\SETUPDRV.STD</SetupDrvPath>
    </TSystemInfo>
  </SystemInfo>
  <TimeWaited>0</TimeWaited>
  <TimeUsed>2406</TimeUsed>

```

Hvis der ikke er nogen fejl eller mangler i requesten, så bliver der returneret 'OK' i `<ErrorCode>` og `<ResponseCode>`.

Hvis der er en fejl på applikationsniveau, vil der blive returneret en kort besked i `<ResponseCode>` som angiver hvad fejlen er.

Følgende response koder kan forekomme i `<ResponseCode>`

'PARTIAL DATA'

'NO DATA'

'REQUEST ERROR'

'NO CHANGE'

'LOCKED'

'DELETED'

'REMARKS'

'CALCULATED'

'NO POSITIONS'

'NOT CALCULATED'

Mens der i `<ResponseDescription>` vil der være en mere uddybende tekst.

`<ErrorCode>` anvendes til fejl som ikke er på applikationsniveau, det kan være fejl i forbindelsen mellem webserveren og VIP, eller fejl som ikke er kendte af VIP.

Hvis der i oprettelses funktioner sendes stamdata i requesten som ikke er oprettet i VIP, vil der komme en 'Error' i Responsekoden, og under den position det gælder, vil der i `<ValidationResult>` stå hvilke stamdata det drejer sig om. Data vil stadig blive indlæst i VIP.

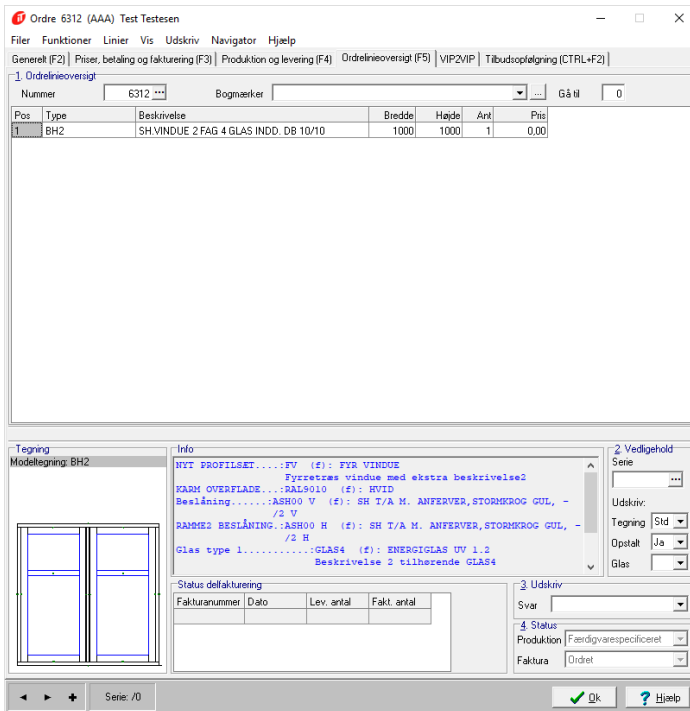
CreateWebOrder/Quote

CreateWebOrder anvendes til at oprette en ordre i VIP, typisk fra et website. I CreateWebOrder kan der oprettes en ordre med model, lager- og tekstpositioner. En modelposition kan oprettes med de samme færdigvaresvar/tvangssvar som der er i VIP.

Eksempel på en request med minimumskrav for oprettelse af en modelposition:

```
<?xml version="1.0" encoding="UTF-16"?>
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Data>
    <TOrderData>
      <Order>
        <TOrder>
          <Customer>
            <Customer>
              <CustomerNo>101003</CustomerNo>
            </Customer>
          </Customer>
          <Position>
            <ModelPosition>
              <Quantity>1</Quantity>
              <Model>BH2</Model>
              <Height>1000</Height>
              <Width>1000</Width>
            </ModelPosition>
          </Position>
        </TOrder>
      </Order>
    </TOrderData>
  </Data>
</Request>
```

Ovenstående request vil oprette en ordre i VIP med én position, med færdigvaresvar fra modellen. Hvis der ønskes andre færdigvaresvar, kan disse indsættes i requesten. Hvis der i requesten findes svar som ikke kan indsættes i færdigvaresvar, vil de blive indsat som tvangssvar.



Tips!

- Ønskes der indsat flere tilbehør på karmen, så oprettes der en ekstra **<FrameClearance>** med **<Index>0</Index>**.

Eksemplet nedenunder vil oprette en position med to tilbehør på karm og ét tilbehør i ramme 1:

```

<?xml version="1.0" encoding="UTF-16"?>
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Data>
    <TOrderData>
      <Order>
        <TOrder>
          <Customer>
            <Customer>
              <CustomerNo>101003</CustomerNo>
            </Customer>
          </Customer>
          <OrderType>AAA</OrderType>
          <Position>
            <ModelPosition>
              <Quantity>1</Quantity>
              <Model>BH2</Model>
              <Height>1000</Height>
              <Width>1000</Width>
            </ModelPosition>
            <Frame>
              <TFrame>
                <Supplement>
                  <TSupplementSpecification>
                    <Index/>
                    <Key>GREB1</Key>
                  </TSupplementSpecification>
                </Supplement>
                <FrameClearance>
                  <TFrameClearance>
                    <Index>0</Index>
                    <Sash>
                      <TSash>
                        <Supplement>
                          <TSupplementSpecification>
                            <Index/>
                            <Key>V20E-VSK</Key>
                          </TSupplementSpecification>
                        </Supplement>
                      </TSash>
                    </Sash>
                  </TFrameClearance>
                </TFrameClearance>
                <TFrameClearance>
                  <Index>1</Index>
                  <Sash>
                    <TSash>
                      <Supplement>
                        <TSupplementSpecification>
                          <Index/>
                          <Key>GREB2</Key>
                        </TSupplementSpecification>
                      </Supplement>
                    </TSash>
                  </Sash>
                </TFrameClearance>
              </FrameClearance>
            </Layer>
          </TFrame>
        </Frame>
      </ModelPosition>
    </Position>
  </TOrder>
</Order>
</TOrderData>
</Data>
</Request>

```

- Hvis der på modellen er henvisning på glasfelter, kan alle glasfelter udfyldes med det samme glas ved at anvende `<Glass>` under `<Frame>`. Det samme er gældende for fyldninger.

```

<Frame>
  <TFrame>
    <Panel>
      <TPanelSpecification>
        <Key/>
      </TPanelSpecification>
    </Panel>
    <Glass>
      <TGlassSpecification>
        <Key/>
        <PanelType/>
        <DeliveredLoose/>
      </TGlassSpecification>
    </Glass>
  </TFrame>
</Frame>

```

- Der kan i VIP oprettes grupper på tilbehør. Dette bevirker, at der fra CreateWebOrder er mulighed for både at tilføje yderligere tilbehør, men også udskifte et tilbehør.

Hvis der i CreateWebOrder indsendes et tilbehør, som findes i samme gruppe som det tilbehør der findes på modellen, og rammenummeret er det samme, så vil tilbehøret fra modellen blive erstattet med det tilbehør der kommer fra CreateWebOrder.

Findes der et tilbehør på modellen som er i samme gruppe som det der er med i CreateWebOrder, men rammenummeret ikke er det samme, så vil tilbehøret fra modellen beholdes, og tilbehøret fra CeateWebOrder oprettes som et tvangssvar.

Hvis der i et rammefelt ønskes at udskifte tilbehøret, så skal tilbehøret være i samme gruppe som det eksisterende, og så er det `<Index>` for `<FrameClearance>` der styrer om det skal indlæses. Her anvendes `<Index>` under `<Supplement>` ikke.

Eksempel på response for CreateWebOrder:

```

<?xml version="1.0" encoding="UTF-16"?>
<Response xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ErrorCode>OK</ErrorCode>
  <ErrorName>OK</ErrorName>
  <ErrorDescription/>
  <ResponseCode/>
  <ResponseDescription/>
  <ResponseCodeGroup>OK</ResponseCodeGroup>
  <ResponseInfo/>
  <SystemInfo>
    <TSystemInfo>
      <Version>17.70.0.T</Version>
      <Revision>25066</Revision>
      <CompileDate>03-09-2019 13:45:19</CompileDate>
      <WindowsUserName>lcg</WindowsUserName>
      <Module>Vip.exe</Module>
      <Database>VIP</Database>
      <SetupDrvPath>\\lgudvfil\VIP\PROG\SETUPDRV.STD</SetupDrvPath>
    </TSystemInfo>
  </SystemInfo>
  <TimeWaited>0</TimeWaited>
  <TimeUsed>388</TimeUsed>
  <Data>
    <TOrderData>
      <Order>
        <TOrder>
          <Number>6287</Number>
          <Date>2019-09-30</Date>
          <Position>
            <ModelPosition>
              <Number>6287</Number>
              <PosNo>1</PosNo>
              <SubPosNO/>
              <ValidationResult>
                <TValidationResult>
                  <Error/>
                  <Warning/>
                </TValidationResult>
              </ValidationResult>
            </ModelPosition>
          </Position>
        </TOrder>
      </Order>
    </TOrderData>
  </Data>
</Response>

```

Samtlige response indeholder et `<Data>` element, hvor relevante data for requesten bliver returneret. I `<Data>` elementet findes `<TValidationResult>`, hvis der kommer en valideringsfejl i forbindelse med kaldet, vil der i dette tag blive returneret information om hvilke data der ikke kan valideres.

```
<?xml version="1.0" encoding="UTF-16"?>
<Response xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ErrorCode>OK</ErrorCode>
  <ErrorName>OK</ErrorName>
  <ErrorDescription/>
  <ResponseCode>Error</ResponseCode>
  <ResponseDescription>Validation error</ResponseDescription>
  <ResponseCodeGroup>Error</ResponseCodeGroup>
  <ResponseInfo/>
  <SystemInfo>
    <TSystemInfo>
      <Version>17.69.25</Version>
      <Revision>26123</Revision>
      <CompileDate>04-11-2019 11:58:64</CompileDate>
      <WindowsUserName>lcg</WindowsUserName>
      <Module>Vip.exe</Module>
      <Database>VIP</Database>
      <SetupDrvPath>\\lgudvfil\VIP\PROG\SETUPDRV.STD</SetupDrvPath>
    </TSystemInfo>
  </SystemInfo>
  <TimeWaited>0</TimeWaited>
  <TimeUsed>507</TimeUsed>
  <Data>
    <TOrderData>
      <Order>
        <TOrder>
          <Number>5776</Number>
          <Date>2019-11-07</Date>
          <Position>
            <ModelPosition>
              <Number>5776</Number>
              <PosNo>1</PosNo>
              <SubPosNO/>
              <ValidationResult>
                <TValidationResult>
                  <Error>
                    <string>Hinging: TEST </string>
                  </Error>
                  <Warning/>
                </TValidationResult>
              </ValidationResult>
            </ModelPosition>
          </Position>
        </TOrder>
      </Order>
    </TOrderData>
  </Data>
</Response>
```

CalcOrder/Quote

CalcOrder/Quote kalkulerer en hel ordre hvis der kun angives et ordrenummer. Ved angivelse af et positionsnummer, er det kun positionen der bliver beregnet.

Der kan vælges mellem kalkulation inkl. pris eller en tegningskalkulation. Den ønskede kalkulation angives med '1'. Hvis der ikke angives noget i kalkulation, vil der blive foretaget en kalkulation inkl. pris, hvor 'Beregn ny rabat' er valgt. Angives der '1' i `<CalculatePrice>` bliver ordren kalkuleret med 'Beregn ny pris'.

```
<?xml version="1.0" encoding="UTF-16"?>
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Number>5773</Number>
  <PosNo>3</PosNo>
  <SubPosNO/>
  <CalculatePrice>1</CalculatePrice>
  <CalculateDrawing/>
</Request>
```

I `<Data>` elementet findes `<Calculation>`, hvis det ikke er muligt at kalkulere ordren, vil der i dette tag blive returneret information om hvilke data der ikke kan kalkuleres.

```

<?xml version="1.0" encoding="UTF-16"?>
<Response xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ErrorCode>OK</ErrorCode>
  <ErrorName>OK</ErrorName>
  <ErrorDescription/>
  <ResponseCode>NOT CALCULATED</ResponseCode>
  <ResponseDescription>One or more positions were calculated with errors for Order: 5775
</ResponseDescription>
  <ResponseCodeGroup>Error</ResponseCodeGroup>
  <ResponseInfo/>
  <Data>
    <TOrderData>
      <Order>
        <TOrder>
          <Number>5775</Number>
          <Customer>
            <Customer/>
          </Customer>
          <Position>
            <ModelPosition>
              <PosNo>1</PosNo>
              <SubPosNO/>
              <Quantity>1</Quantity>
              <CostPrice>0.00</CostPrice>
              <ListPrice>0.00</ListPrice>
              <DiscountPercent>0.00</DiscountPercent>
              <DiscountAmount>0.00</DiscountAmount>
              <SalesPrice>0.00</SalesPrice>
              <PriceDetails/>
              <Calculation>
                <TCalculation>
                  <CalcResult>CalcDefault</CalcResult>
                  <CalcMessage>
                    <string>Ordre nummer 5775 1 kan ikke beregnes (BH1-KLÆBESPROSSER)</string>
                    <string>Beslag: SH00,SH001 kan ikke nå</string>
                    <string> AR13 Karmfelt:1 Falsbredde:944.0 Falshøjde:944.0</string>
                  </CalcMessage>
                </TCalculation>
              </Calculation>
              <AlternativePrice>False</AlternativePrice>
              <CalculationStatus>0</CalculationStatus>
            </ModelPosition>
          </Position>
          <Economy/>
        </TOrder>
      </Order>
    </TOrderData>
  </Data>
</Response>

```

GetOrder/Quote

Generelt for alle response der henter ordrehoveder er, alle debitorinformationer hentes fra ordrehovedet, ikke fra stamregistret. Hvis der ønskes at hente debitorinformationer fra stamregistret, skal MasterDataMaintenance anvendes.

GetOrder/Quote henter alle informationer på en ordre, både ordrehoved og positioner inkl. alle færdigvaresvar.

Responen fra en GetOrder kan anvendes direkte i en CreateWebOrder. Hvis tagget `<Response>` udskiftes med `<Request>` og enten `<DeliveryDate>` eller `<DeliveryWeek>` fjernes, da disse to tags ikke kan være udfyldt på én gang i CreateWebOrder.

Er man kun interesseret i at hente ordrehoveder, så kan der hentes ét ordrehoved ad gangen med GetOrderHead, ønskes der mere end ét så kan funktionen GetOrderList anvendes.

Det samme gør sig gældende for positioner, med GetSpecificOrderLineList kan der hentes en eller flere positioner ad gangen.

I requesten angives ordre- og positionsnummer.

```
<?xml version="1.0" encoding="UTF-16"?>
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Number>6286</Number>
  <Position>
    <TPosition>
      <PosNo>3</PosNo>
      <SubPosNo/>
    </TPosition>
  </Position>
</Request>
```

Ønskes der flere positioner retur, kopieres taggene under `<TPosition>`


```
<?xml version="1.0" encoding="UTF-16"?>
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Number>6286</Number>
  <Position>
    <TPosition>
      <PosNo>3</PosNo>
      <SubPosNo/>
    </TPosition>
    <TPosition>
      <PosNo>5</PosNo>
      <SubPosNo/>
    </TPosition>
  </Position>
</Request>
```

DeleteOrder/Quote

Med Standard API er der mulighed for at slette hele ordrer og positioner.

Funktionen DeleteOrder sletter en hel ordre:

```
<?xml version="1.0" encoding="UTF-16"?>
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Number/>
</Request>
```

Med DeleteOrders er der mulighed for at slette flere ordre på én gang ud fra de valgte kriterier.

```
<?xml version="1.0" encoding="UTF-16"?>
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Number/>
  <RevisionDateFrom/>
  <RevisionDateTo/>
  <CreateDateFrom/>
  <CreateDateTo/>
  <Flow/>
  <FlowStatusFrom/>
  <FlowStatusTo/>
  <StatusChangeFrom/>
  <StatusChangeTo/>
  <OrderTypeFrom/>
  <OrderTypeTo/>
  <OnlyOrder/>
  <ChangeDateStatusFrom/>
  <ChangeDateStatusTo/>
  <NumberFrom/>
  <NumberTo/>
</Request>
```

```
<DebNumberFrom/>
<DebNumberTo/>
<DeliveryDateFrom/>
<DeliveryDateTo/>
<RelativeDate/>
</Request>
```

Ønskes der kun at slette én enkelt position anvendes DeleteOrderLine

```
<?xml version="1.0" encoding="UTF-16"?>
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Number/>
  <PosNo/>
  <SubPosNo/>
</Request>
```

Skal der slettes flere positioner på én gang anvendes DeleteOrderLines på samme måde som GetSpecificOrderLineList, hvor taggene under <TPosition> kopieres

```
<?xml version="1.0" encoding="UTF-16"?>
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Number>6286</Number>
  <Position>
    <TPosition>
      <PosNo>3</PosNo>
      <SubPosNo/>
    </TPosition>
    <TPosition>
      <PosNo>5</PosNo>
      <SubPosNo/>
    </TPosition>
  </Position>
</Request>
```

GetRevisedOrderList/QuoteList

Med funktionerne GetRevisedXXX kan der hentes lister på ændrede ordrer. Requesten til funktionerne er bygget ens op med henblik på anvendelse af Rowversion (se side 3). I `<MaxRecordCount/>` angives hvor mange informationer der ønskes i responsen, hvis der ikke angives en værdi, vil der blive returneret 10 svar ad gangen. I tagget `<IncludeDeleted>` kan der angives hvorvidt der ønskes at få slettede ordrehoveder og positioner med retur. Tagget skal angives med enten 'true' eller '1' for at medtage slettede ordrehoveder og positioner.

GetRevisedOrderHead/QuoteHeadList henter alle informationer på ordrehovedet

```
<?xml version="1.0" encoding="UTF-16"?>
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <FromRowVersion>7084200</FromRowVersion>
  <MaxRecordCount/>
  <IncludeDeleted>1</IncludeDeleted>
</Request>
```

GetRevisedOrder/QuoteList henter alle informationer på ordrehovedet samt positioner

```
<?xml version="1.0" encoding="UTF-16"?>
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <FromRowVersion>7084200</FromRowVersion>
  <MaxRecordCount/>
  <IncludeDeleted>1</IncludeDeleted>
</Request>
```

GetRevisedOrderLine/QuoteLineList henter alle informationer på positioner

```
<?xml version="1.0" encoding="UTF-16"?>
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <FromRowVersion>7084200</FromRowVersion>
  <MaxRecordCount/>
  <IncludeDeleted>1</IncludeDeleted>
</Request>
```

CalcModel

CalcModel kan anvendes med henblik på at hente en pris.

Eksempel på en request med minimumskrav:

```
<?xml version="1.0" encoding="UTF-16"?>
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Data>
    <ModelPosition>
      <Quantity>1</Quantity>
      <Model>BH2</Model>
      <Height>1000</Height>
      <Width>1000</Width>
    </ModelPosition>
  </Data>
</Request>
```

På samme måde som CreateWebOrder kan der udskiftes færdigvaresvar fra dem modellen er oprettet med og til de ønskede svar, ved at indsætte dem i requesten.

GetOrderPrint/QuotePrint

GetOrderPrint returnere en ordre som PDF.

```
<?xml version="1.0" encoding="UTF-16"?>
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Number/>
  <PrintNoPrice/>
</Request>
```

Ønskes der ikke at prisen skal medtages på PDF'en, udfyldes `<PrintNoPrice/>` med '1' eller 'True'. Såfremt der ønskes en pris med, skal tagget udfyldes med '0' eller 'False', eller blot efterlades blank.

GetOrderPositionPrintImage

Henter et billede af modellen på den angivne position.

Der kan vælges følgende billede formater: GIF, PNG, JPG (JPEG). Hvis der ikke angives et format, vil outputtet blive genereret som en metafile.

Ønskes karm- og feltnr, målsætning, beslåning osv medtaget, så skal der angives '1' eller 'true' i taggene.

```
<?xml version="1.0" encoding="UTF-16"?>
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Number/>
  <PosNo/>
  <SubPosNO/>
  <Parameters>
    <TDrawingParameters>
      <FrameNumber/>
      <FrameClearanceNumber/>
      <Supplement/>
      <Hinging/>
      <Measurement/>
      <ImageType/>
      <MaxImageWidth/>
      <MaxImageHeight/>
    </TDrawingParameters>
  </Parameters>
</Request>
```

UpDateOrderHead

Med funktionen UpDateOrderHead er det muligt at opdatere udvalgte felter på ordrehovedet. Angiv det ordrenummer, der skal opdateres og medtag de tag, der indeholder de felter og værdi, der ønskes opdateret på ordren.

Det er ikke muligt at ændre debitor eller seriesætte en ordre med denne funktion.

```
<?xml version="1.0" encoding="UTF-16"?>
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Data>
    <TOrder>
      <Number>20284</Number>
      <Date xsi:nil="true"/>
      <Attention>Jens Grøn</Attention>
      <DeliveryInfo1>Husk at ringe til kunden 5 timer før levering</DeliveryInfo1>
      <DeliveryInfo2 xsi:nil="false"/>
      <QuoteNumber xsi:nil="true"/>
      <RevisionInIt xsi:nil="true"/>
    </TOrder>
  </Data>
</Request>
```

WorkflowInput

Med workflowinput er det muligt at ændre status og/eller knytte et notat på en ordre i Workflow.

```
<?xml version="1.0" encoding="UTF-16"?>
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Number/>
  <Input/>
  <Notat/>
  <WorkCenter/>
</Request>
```

`<Input/>` udfyldes med det 'Flow input' der er oprettet i VIP til Standard API

Notatet bliver tilknyttet det arbejdscenter der bliver angivet i `<WorkCenter/>`.

Get/SetUserDefinedFields

I Vip er der mulighed for at oprette egne felter (oprettes i VIPBase) i stamregistre. Værdien i felterne kan hentes og vedligeholdes via Standard API.

Værdierne fra felterne kan hentes enten via MasterDataMaintenance.

```
<?xml version="1.0" encoding="UTF-16"?>
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <RequestOperation>R</RequestOperation>
  <Data>
    <Customer>
      <CustomerNo>004989564589</CustomerNo>
    </Customer>
  </Data>
</Request>
```

The screenshot shows the Integrationstest interface with the following details:

- Forbindelse:** Embedded
- Grp.:** MasterData
- Funktion:** MasterDataMaintenance
- Version:** 2
- URL:** win7unittest.intranet.logodan.dk
- Request (Left Panel):**

```
<?xml version="1.0" encoding="UTF-16"?>
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <RequestOperation>R</RequestOperation>
  <Data>
    <Customer>
      <CustomerNo>004989564589</CustomerNo>
    </Customer>
  </Data>
</Request>
```
- Response (Right Panel):**

```
</TSystemInfo>
</SystemInfo>
<TimeWaited>0</TimeWaited>
<TimeUsed>62</TimeUsed>
<Data>
  <Customer>
    <UserDefined>
      <UserDefined>
        <Name>Bonus</Name>
        <Value>True</Value>
      </UserDefined>
    </UserDefined>
  </Customer>
  <CustomerNo>004989564589</CustomerNo>
```
- Stack:** 224052
- Gennemløbstid:** 149

eller via GetUserDefinedFields, som henter dem ønskede værdi:

```
<?xml version="1.0" encoding="UTF-16"?>
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Fields>
    <TUserDefinedField>
      <Entity>T_Debitor</Entity>
      <Key>004989564589</Key>
      <Name>Bonus</Name>
    </TUserDefinedField>
  </Fields>
</Request>
```

The screenshot shows the Integrationstest interface with the following details:

- Forbindelse:** Embedded
- Grp.:** MasterData
- Funktion:** GetUserDefinedFields
- Version:** 2
- URL:** win7unittest.intranet.logodan.dk
- Request (Left Panel):**

```
<?xml version="1.0" encoding="UTF-16"?>
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Fields>
    <TUserDefinedField>
      <Entity>T_Debitor</Entity>
      <Key>004989564589</Key>
      <Name>Bonus</Name>
    </TUserDefinedField>
  </Fields>
</Request>
```
- Response (Right Panel):**

```
<?xml version="1.0" encoding="UTF-16"?>
<Response xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ErrorCode>OK</ErrorCode>
  <ErrorName>OK</ErrorName>
  <ErrorDescription/>
  <ResponseCode/>
  <ResponseDescription/>
  <ResponseCodeGroup>OK</ResponseCodeGroup>
  <ResponseInfo/>
  <TimeWaited>0</TimeWaited>
  <TimeUsed>16</TimeUsed>
  <Fields>
    <TUserDefinedField>
      <Entity>T_Debitor</Entity>
      <Key>004989564589</Key>
      <Name>Bonus</Name>
      <Value>True</Value>
    </TUserDefinedField>
```
- Stack:** 224052
- Gennemløbstid:** 65

Værdien kan på samme måde vedligeholdes via SetUserDefinedFields med følgende kald:

```
<?xml version="1.0" encoding="UTF-16"?>
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Fields>
    <TUserDefinedField>
      <Entity>T_Debitor</Entity>
      <Key>004989564589</Key>
      <Name>Bonus</Name>
      <Value>False</Value>
    </TUserDefinedField>
  </Fields>
</Request>
```

Værdien kan fjernes helt, ved at udelade `<Value>` fra kaldet, eller sætte den til NULL:

```
<?xml version="1.0" encoding="UTF-16"?>
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Fields>
    <TUserDefinedField>
      <Entity>T_Debitor</Entity>
      <Key>004989564589</Key>
      <Name>Bonus</Name>
      <Value xsi:nil="true"/>
    </TUserDefinedField>
  </Fields>
</Request>
```

Både `SetUserDefinedFields` og `GetUserDefinedFields` kan behandle flere deltransaktioner i samme kald, eksempelvis følgende kald på `GetUserDefinedFields` indeholdende tre deltransaktioner:


```
<?xml version="1.0" encoding="UTF-16"?>
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Fields>
    <TUserDefinedField>
      <Entity>T_Debitor</Entity>
      <Key>004989564589</Key>
      <Name>Bonus</Name>
    </TUserDefinedField>

    <TUserDefinedField>
      <Entity>T_Debitor</Entity>
      <Key>004989564589</Key>
      <Name>Kommentar</Name>
    </TUserDefinedField>

    <TUserDefinedField>
      <Entity>T_Test</Entity>
      <Key>1234</Key>
      <Name>Tester</Name>
    </TUserDefinedField>
  </Fields>
</Request>
```

```
<?xml version="1.0" encoding="UTF-16"?>
<Response xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ErrorCode>OK</ErrorCode>
  <ErrorName>OK</ErrorName>
  <ErrorDescription/>
  <ResponseCode>PartialOk</ResponseCode>
  <ResponseDescription>Some data is invalid. Please see FieldsWithError.</ResponseDescription>
  <ResponseCodeGroup>Remark</ResponseCodeGroup>
  <ResponseInfo/>
  <Fields>
    <TUserDefinedField>
      <Entity>T_Debitor</Entity>
      <Key>004989564589</Key>
      <Name>Bonus</Name>
      <Value>False</Value>
    </TUserDefinedField>
    <TUserDefinedField>
      <Entity>T_Debitor</Entity>
      <Key>004989564589</Key>
      <Name>Kommentar</Name>
      <Value/>
    </TUserDefinedField>
  </Fields>
  <FieldsWithError>
    <TUserDefinedFieldWithError>
      <Entity>T_Test</Entity>
      <Key>1234</Key>
      <Name>Tester</Name>
      <Error>Entity not found</Error>
    </TUserDefinedFieldWithError>
  </FieldsWithError>
</Response>
```

Såfremt der opstår fejl i en deltransaktion, sættes **<ResponseCode>** til "PartialOk" og **<ResponseCodeGroup>** til "Remark" samt der inkluderes en **<Error>** med en fejlbesked på den pågældende deltransaktion.

Ovenstående resulterer i to gyldige svar i `<Fields>` samt én fejlmeddelelse i `<FieldsWithError>`, da der forsøges at hente oplysning fra en ugyldig Entity (T_Test):

Vær opmærksom på, såfremt der spørges på en værdi, der endnu ikke findes for det pågældende brugerdefinerede felt for objektet, forsøges returneret en standardværdi svarende til værdien brugeren vil opleve i GUI. Streng vil derfor være tomme, integers/real være 0 og booleans sat til False.

Ved vedligeholdelse af feltværdier via MasterDataMaintenance er det vigtigt at være opmærksom på, at brugerdefinerede felter kun påvirkes af ændringen, hvis de er medtages i kaldet. Følgende kald vil derfor både nulstille værdien for "Bonus" (derefter implicit sættes den til False, da det er en boolean) samt tilknytte en kommentar til objektet:

```
<?xml version="1.0" encoding="UTF-16"?>
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <RequestOperation>U</RequestOperation>
  <Data>
    <Customer>
      <CustomerNo>004989564589</CustomerNo>
      <UserDefined>
        <TUserDefined>
          <Name>Bonus</Name>
          <Value xsi:nil="true"/>
        </TUserDefined>
        <TUserDefined>
          <Name>Kommentar</Name>
          <Value>Ny kommentar</Value>
        </TUserDefined>
      </UserDefined>
    </Customer>
  </Data>
</Request>
```

Hvorimod nedenstående kald nulstiller ikke "Bonus", men kun opdaterer kommentaren:

```
<?xml version="1.0" encoding="UTF-16"?>
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <RequestOperation>U</RequestOperation>
  <Data>
    <Customer>
      <CustomerNo>004989564589</CustomerNo>
      <UserDefined>
        <TUserDefined>
          <Name>Kommentar</Name>
          <Value>Opdateret kommentar</Value>
        </TUserDefined>
      </UserDefined>
    </Customer>
  </Data>
</Request>
```

Ved læsning af objektet via `<RequestOperation>R</RequestOperation>`

vil alle tilgængelige værdier dog blive returneret, hvorfor det ikke er nødvendigt/muligt at specificere hvilke brugerdefinerede felter man ønsker returneret via forespørgslen. GetUserDefinedFields skal anvendes i sådanne tilfælde

Gyldige værdier er:

Klasse (<Entity>)	Felt navn(<Name>)	Felttype	Ledetekst	Feltlængde
- T_Ordrehoved - T_Tilbudhoved - T_Debitor - T_Ordrelinje - T_Tilbudlinje - T_Beslaaning - T_Glas - T_KonstKarm - T_KonstModel - T_Overflade - T_Profiler - T_Tilbehor - T_TvangsSvarHoved	A-Z, a-z, 0-9. Skal være unik.	Alle typer nedarvet fra TBasisInteger, TBasisReal, TBasisString og TBasisBoolean kan anvendes, inkl. Integer, Real, Boolean og String.	Minimum ét tegn.	0 eller positivt tal. Angives 0 bruges typens standardlængde.

MasterDataMaintenance

Med MasterDataMaintenance er der mulighed for at vedligeholde stamregistre.

Der kan skabes nye poster, ændres i poster, hente data på en specifik post eller der kan hentes en liste af poster. I [<RequestOperation/>](#) angives den ønskede operation:

U – Update

Opdaterer en række i det valgte register med de tags der er udfyldt i requesten. Tags der sendes med og som ikke er udfyldt vil fjerne data der findes i registeret.

C – Create

Opretter en ny række i det valgte register.

Kombinationen **UC**, vil opdatere en række, såfremt nøglen allerede findes, ellers vil der blive oprettet en ny post.

L – List

Returnere en liste over det eller de registre der sendes med i requesten.

R - Read

Returnere en enkelt række, nøglen skal være udfyldt.

D-Delete

Sletter rækken i det valgte register

RL-RevisedList

Anvendes sammen med <FromRowVersion>. Der kan hentes data fra alle registre ud fra rowversion.

LD-DeletedList

Anvendes sammen med <FromRowVersion>. Der kan hentes slettede nøgler samt datoen for hvornår sletningen er foretaget, ud fra rowversion og <DeletedItems> klassen.

Opdatering af beslag:

```
<?xml version="1.0" encoding="UTF-16"?>
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <RequestOperation>U</RequestOperation>
  <FromRowVersion/>
  <MaxRecordCount/>
  <Data>
    <Hinging>
      <Key>ASH-VÆGT1</Key>
      <Description>Test Beslag</Description>
    </Hinging>
  </Data>
</Request>
```

Hentning af information på en ramme:

```
<?xml version="1.0" encoding="UTF-16"?>
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <RequestOperation>R</RequestOperation>
  <FromRowVersion/>
  <MaxRecordCount/>
  <Data>
    <Sash>
      <Key>R11</Key>
    </Sash>
  </Data>
</Request>
```

Ønskes der at fjerne data i stamregistre, skal tagget medsendes tomt, hvis eksisterende data i stamregistre ønskes bibeholdt i et felt, så skal tagget fjernes fra requesten.

GetProductionData

Med GetProductionData kan der hentes en råvareliste.

```
<?xml version="1.0" encoding="UTF-16"?>
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <MainBatchNo/>
  <BatchNoFrom/>
  <BatchNoTo/>
  <Number/>
  <PosNo/>
  <BatchPosNoFrom/>
  <BatchPosNoTo/>
  <CriteriaInclude/>
  <Criteria1/>
  <Criteria2/>
  <Criteria3/>
  <Criteria4/>
  <Criteria5/>
</Request>
```

Disse fem kriteriefelter er fra lagerregisteret og kan her bruges til filtrering af lagervarerne. Der må kun anvendes tal eller bogstaver, rækkefølgen er underordnet, men der skelnes mellem store og små bogstaver.

C#

Eksempel på anvendelse af funktionen "GetOrder_V2" i VIP Standard API. Funktionen anvendes til at hente ordredata for en specifik ordre i VIP. I dette eksempel efterspørges data på ordre 3049:

```
Request aRequest = new Request();
Response aResponse = new Response();
VipWebServiceClient service = new VipWebServiceClient();
XmlSerializer serializer = new XmlSerializer(typeof(Request));
XmlSerializer deserializer = new XmlSerializer(typeof(Response));
StringWriter stringWriter = new StringWriter();
var writer = XmlWriter.Create(stringWriter);

aRequest.Number = "3049";
serializer.Serialize(writer, aRequest);
string serializedXML = stringWriter.ToString();

string ResponseXML = service.Request("GetOrder_V2", serializedXML);
Console.WriteLine("Response: " + ResponseXML);

StringReader stringReader = new StringReader(ResponseXML);
aResponse = (Response)deserializer.Deserialize(stringReader);
```