

10. RFC ubl-invoice, slashes

25mrt22: From: c.graumans@chello.nl <c.graumans@chello.nl>

To: Hoogenberg, J. (Jan) <Jan.Hoogenberg@frieslandcampina.com>

Subject: RE: eDAIRY ubl-invoice met json tags eromheen

Jan, Ik begrijp de xml zoals die nu uit jullie systeem komt niet.

Waar komen de backslashes \" en \ vandaan? Die horen niet thuis in een xml-bericht.

Ik de RFC xml staat bijvoorbeeld:

```
<n2:Invoice xmlns:n2=\"urn:oasis:names:specification:ubl:schema:xsd:Invoice-2\" xmlns:cac=\"urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2\" xmlns:xsi=\"http://www.oasis-open.org/ubl/os-UBL-2.1/xsd/maindoc/UBL-Invoice-2.1.xsd\"><cbc:UBLVersionID></cbc:UBLVersionID><cbc:ProfileID></cbc:ProfileID><cbc:ID>0092002494</cbc:ID><cbc:IssueDate>2018-01-31</cbc:IssueDate><cbc:DueDate></cbc:DueDate><cbc:Note></cbc:Note><cbc:DocumentCurrencyCode listID=\"ISO 4217 Alpha\"></cbc:DocumentCurrencyCode><cbc:DescriptionCode listID=\"CL???\">2018010</cbc:DescriptionCode></cac:InvoicePeriod><cac:Attachment><EmbeddedDocumentBinaryObject mimeType=\"application/pdf\" filename=\"Melkgeld...></EmbeddedDocumentBinaryObject></cac:Attachment><cac:AccountingSupplierParty><cbc:CustomerAssignedAccountID schemeID=\"nl.plnt\">001780250</cbc:CustomerAssignedAccountID><cbc:PartyIdentification><ID schemeID=\"nl.kvk\">32164040</ID></cbc:PartyIdentification></cac:AccountingSupplierParty><cbc:StreetName></cbc:StreetName><cbc:CityName>LEUSDEN</cbc:CityName><cbc:PostalZone>3831 SB</cbc:PostalZone><cbc:PartyTaxScheme><cbc:RegistrationName>HG van Burgsteden</cbc:RegistrationName><cbc:CompanyRegistrationName></cbc:CompanyRegistrationName></cac:PartyTaxScheme></n2:Invoice>
```

Het moet er ongeveer zo uitzien (ik heb een gecorrigeerd voorbeeld bijgevoegd):

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- eDAIRY test message RFC, modified by Conny graumans 26jan21 -->
<n2:Invoice xmlns:n2="urn:oasis:names:specification:ubl:schema:xsd:Invoice-2" xmlns:cac="urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <cbc:UBLVersionID>UBL v2.1</cbc:UBLVersionID>
  <cbc:CustomizationID>UBL-invoice-v2.1-eDAIRY</cbc:CustomizationID>
  <cbc:ProfileID>UBL-invoice-v2.1-eDAIRY</cbc:ProfileID>
  <cbc:ID>0092093766</cbc:ID>
  <cbc:IssueDate>2018-07-31</cbc:IssueDate>
  <cbc:DueDate>2018-08-08</cbc:DueDate>
  <cbc:InvoiceTypeCode listID="UNCL1001">389</cbc:InvoiceTypeCode>
  <cbc:Note languageID="NL">Afrekening melkgeld juli 2018</cbc:Note>
  <cbc:DocumentCurrencyCode listID="ISO 4217 Alpha">EUR</cbc:DocumentCurrencyCode>
  <cac:InvoicePeriod>
    <cbc:StartDate>2018-07-01</cbc:StartDate>
    <cbc:EndDate>2018-07-31</cbc:EndDate>
    <cbc:DescriptionCode listID="CL???">2018070</cbc:DescriptionCode>
  </cac:InvoicePeriod>
  <cac:AdditionalDocumentReference>
    <cbc:ID>AB_MGA-0092093766</cbc:ID>
  </cac:AdditionalDocumentReference>
</n2:Invoice>
```

Dat kan veroorzaakt zijn doordat je de xml als txt hebt weggeschreven waarbij iedere einde regel door een \ is vervangen (?) Ik heb de overbodige \ eruit gehaald in bijgevoegd voorbeeld.

Verder kwam ik in het ubl-invoice xml voorbeeld (onderdeel van de aangeleverde Json array) de volgende fouten tegen die hersteld moeten worden:

<cbc:UBLVersionID>123</cbc:UBLVersionID> moet ingevuld zijn

<ID schemeID="nl.kvk"> moet zijn <cbc:ID schemeID="nl.kvk">

<EmbeddedDocumentBinaryObject mimeType="application/pdf" moet zijn

<cbc:EmbeddedDocumentBinaryObject mimeType="application/pdf"

</EmbeddedDocumentBinaryObject> moet zijn </cbc:EmbeddedDocumentBinaryObject>

<ID schemeID="nl.plnt">001</cbc:ID> moet zijn <cbc:ID schemeID="nl.plnt">001</cbc:ID>

<cac:GoodsItem> moet zijn <cac:Item>

![CDATA[weglaten, moet zijn

filename="Melkgeldnota_201801_0000780250_20180208_135652.pdf">JVBERi0xLjMNCiXi48TDQoIUINUWF etc. etc... Zie bijgevoegd voorbeeld.

`mimeCode="application/pdf" filename="Melkgeldnota_201801_0000780250_20180208_135652.pdf">` moet zijn
(volgorde omdraaien) `filename="Melkgeldnota_201801_0000780250_20180208_135652.pdf"`
`mimeCode="application/pdf">`

`<cbc:ActualDeliveryDate/>` datum invullen plus eindtag `<cbc:ActualDeliveryDate>2018-07-01</cbc:ActualDeliveryDate>`

`<TaxSubtotal>` moet zijn `<cac:TaxSubtotal>`

Graag bereid tot nader overleg, Groet, Conny Graumans, AgroConnect

29mrt22: Van: Hoogenberg, J. (Jan) <Jan.Hoogenberg@frieslandcampina.com> Aan: c.graumans@chello.nl
Conny, De backslashes (\ " en \ ") zijn nodig om dit in json in te bedden (escape characters), zonder deze krijg je syntax fouten in de json. Ik heb ook de versie van JoinData opgevraagd en deze is hetzelfde als die wij nu hebben. Misschien goed als ik eens schakel met VanAaken om te kijken hoe zij dit zien, is tenslotte belangrijkste developer van onze klanten.

29mrt22: Conny aan Jan:
...en hoe krijgt de ontvanger die slashes er dan weer uit; inclusief de slashes is de xml niet valide.

29mrt22: Van: Hoogenberg, J. (Jan) <Jan.Hoogenberg@frieslandcampina.com> Aan: c.graumans@chello.nl
Ik heb zojuist met Jan Herrygers gesproken, volgens hem kan hij dat handmatig met XMLSpy doen door de JSON in te lezen en dan wordt deze gede-escaped. Klonk als niet ingewikkeld, maar ik heb er geen verstand van. Tevens geeft hij aan dat als het handmatig met XMLSpy kan dan moet het met andere tooling ook automatisch kunnen.

Hij stuurde de volgende link: <https://www.altova.com/json-tools>

Verder vond ik nog het volgende op internet:

Escape XML in JSON - JSON String Escape / Unescape

- Backspace is replaced with \b.
- Form feed is replaced with \f.
- Newline is replaced with \n.
- Carriage return is replaced with \r.
- Tab is replaced with \t.
- Double quote is replaced with \"
- Backslash is replaced with \\

29mrt22, Van: c.graumans@chello.nl <c.graumans@chello.nl> Aan: 'Hoogenberg, J. (Jan)' <Jan.Hoogenberg@frieslandcampina.com>

...maar Jan, dat moet je toch niet willen dat vanwege JoinData iedereen zijn xml ubl-invoice moet opschonen; of wordt dat dan weer een extra service die JoinData aanbiedt?

30mrt22: uit telefonisch overleg met Jan Herrygers (VAA).

Is een keuze van JoinData om alles in Json te doen. Betekent inderdaad dat alle xml-berichten die via JoinData uitgevraagd worden, worden vertaald naar Json varianten waarbij de noodzakelijke escape-characters (backslash combinaties) in het bericht worden ingevoerd. De ontvangende partij ontvangt dan geen xml maar een Jason variant daarop. Mocht de ontvangende partij dat weer terug willen vertalen naar Json dan dient deze de escape-characters weer te verwijderen.

Conclusie Conny Graumans: Het is vreemd dat een intermediaire partij (JoinData) de berichten inhoudelijk aanpast en xml-berichten vertaalt naar een Json variant daar waar de ontvangende partij eigenlijk xml dacht te ontvangen. De vraag is of je de xml ubl-invoices wel via JoinData zou moeten willen uitwisselen; er zijn alternatieve wegen die de bericht syntax ongemoeid laten en netjes xml doorleveren.

31mrt22: Van: c.graumans@chello.nl <c.graumans@chello.nl> Aan: 'Gijs@join-data.nl' <Gijs@join-data.nl>
CC: Gerrit Smeenk (FrieslandCampina) <Gerrit.Smeenk@frieslandcampina.com>

Beste Gijs, Zie bijgevoegde issuelog. Kun je aangeven wat hier precies gebeurt?

Is het inderdaad zou dat wanneer partijen via JoinData de xml ubl-edairy-invoice opvragen een Json variant krijgen aangeleverd?

6apr22: Van: Gijs Nijman <Gijs@join-data.nl> Aan: c.graumans@chello.nl
CC: Gerrit Smeenk (FrieslandCampina) <Gerrit.Smeenk@frieslandcampina.com>

Ja, we gooien er een json wrapper omheen (de joindata envelop). Dat doen we omdat er meta data toegevoegd kan worden en meerdere bronnen benaderd kunnen worden. Om een XML bericht in een JSON bericht te doen dient deze volgens, in overeenstemming met de JSON standaard geescaped te worden.

Elke tool die JSON inleest zou dit automatisch moeten de-escapen. Dit is niet anders dan al onze huidige api's. Ook de XML-leverEAB en XML-FaktuurEAB worden op dezelfde manier aangeboden. Hopende hiermee voldoende antwoord te hebben gegeven.

7apr22: Van: c.graumans@chello.nl <c.graumans@chello.nl> Aan: 'Gijs Nijman' <Gijs@join-data.nl>
CC: 'Gerrit Smeenk (FrieslandCampina)' <Gerrit.Smeenk@frieslandcampina.com>;
'jan.hoogenberg@frieslandcampina.com' <jan.hoogenberg@frieslandcampina.com>
OK Gijs, Kun je garanderen dat met deze conversie van xml -> json -> xml er geen risico is dat data in het bericht verandert?

7apr22: Gijs: Ja, dat kunnen we garanderen.

Status 7apr22: Deze issue in het volgende overleg van de OC aan de orde stellen.