

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Ly Tempel 155029IABB

# **E-ARVETE RAKENDUS ETTEVÕTTE ERPLY SISENDI PÕHJAL**

Bakalaureusetöö

Juhendajad: Inna Švartsman  
MSc  
Sandra Pent

Tallinn 2019

## **Autorideklaratsioon**

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Ly Tempel

20.05.2019

## **Annotatsioon**

Käesoleva töö eesmärgiks on luua esimene versioon E-arvete rakendusest ettevõtte Erply sisendi põhjal. E-arvete rakendus võimaldab kasutajal mugavalt koostada e-arveid, kuid samas ka olemasolevaid e-arveid rakendusse üles laadida ning neid PDF-kujul vaadata.

E-arvete koostamise puhul on väga oluline jälgida, et need vastaksid rahandusministri 11.04.2017 määruse nr 24 „Masintöödeldava algdokumendi juhendi kehtestamine“ lisas „E-arvete juhend“ [1] väljatoodud Eesti e-arve standardile. Selleks tuli autoril põhjalikult eelnevalt mainitud dokumenti analüüsida, et mõista millised nõuded on Eestis e-arvetele kehtestatud.

Lisaks tuli autoril teha endale selgeks, mida üldse e-arved tähendavad, kes on e-arvete operaatorid ning töö koostamise käigus omandada uusi tehnoloogiaid, näiteks raamistikud Vue.js ja Nuxt.js.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 44 leheküljel, 5 peatükki, 47 joonist, 1 tabelit.

## **Abstract**

### **E-Invoicing Application Based on the Input by Erply**

There are a lot of invoices sent and received every single day and most of them in Estonia are in paper or PDF-format. Entering data to accounting systems from paper and PDF invoices is yet time-consuming, error-prone, a waste of natural resources and may be considered as conducting the same work twice. The reasonable solution to solve all these problems is to use e-invoices.

Unfortunately, e-invoices are not that commonly used in Estonia, but as from 1st July 2019 Estonian public sector accepts e-invoices only, private sector companies also must be able to create e-invoices when they need to deliver bills to state-owned companies. Moreover, utilization of e-invoices would be beneficial to private sector, too, as e-invoices help to lessen the work of accountants, because instead of typing invoice information multiple times, it needs to be done only once.

The main aim of the thesis is to develop the first version of E-Invoicing Application on the input by Erply that allows user to both create new invoices and view received invoices in PDF-format and upload them to the database.

When creating e-invoices in Estonia it is essential to follow the Estonian e-invoice standard [1]. To not violate the Estonian e-invoice standard, the author of the thesis had to thoroughly work through previously mentioned document to understand the requirements that are applied to the e-invoices in Estonia.

Furthermore, the author needed to examine what exactly e-invoices are, who are e-invoicing operators and acquire new technologies during developing the application, such as Vue.js, Nuxt.js and Actual Reports.

The thesis is in Estonian language and contains 44 pages of text, 5 chapters, 47 figures, 1 table.

## Lühendite ja mõistete sõnastik

API	<i>Application Programming Interface</i> - rakendusliides
Base64	Kodeerimise ja dekodeerimise meetod, mis teisendab binaarandmed <i>American Standard for Information Interchange</i> (ASCII) tekstiformaati ja vastupidi [2]
CLI	<i>Command Line Interface</i> – käsureprogramm, mis viib läbi erinevaid funktsioone sisestatud teksti põhjal [3]
CSS	<i>Cascading Style Sheets</i> – märgistuskeel HTML dokumentide kujundamiseks
DVK	Dokumendivahetuskeskus
HTML	<i>Hyper Text Markup Language</i> – hüperteksti märgistuskeel ehk keel veebilehtede loomiseks
HTTP	<i>Hypertext Transfer Protocol</i> – hüperteksti edastusprotokoll
ISO	<i>International Organization for Standardization</i> – Rahvusvaheline Standardimisorganisatsioon
JSON	<i>JavaScript Object Notation</i> – JavaScriptil põhinev andmeedastusformaad
PDF	<i>Portable Document Format</i> – elektrooniliste dokumentide formaad
PWA	<i>Progressive Web Application</i> – progressiivne veebirakendus
SPA	<i>Single Page Application</i> – üheleherakendus
SSR	<i>Server-Side Rendering</i> – serveripoolne renderdamine
UI	<i>User Interface</i> – kasutajaliides
UX	<i>User Experience</i> – kasutajakogemus
XML	<i>Extensible Markup Language</i> – laiendatav märgistuskeel
XSD	<i>XML Schema Definition</i> – XML skeemi definitsioon

# Sisukord

1 Sissejuhatus .....	11
1.1 Taust ja probleem .....	11
1.2 Eesmärk .....	12
1.3 Ülevaade tööst .....	12
2 E-arved.....	13
2.1 E-arvete vastuvõtjate register .....	13
2.2 Eesti e-arve standard.....	14
2.2.1 Header.....	15
2.2.2 Invoice .....	16
2.2.3 Footer.....	20
2.3 E-arvete operaatorid Eestis .....	21
2.3.1 Arved.ee, Fitek Premium (Fitek).....	21
2.3.2 Arvekeskus (Omniva).....	22
2.3.3 Amphora e-arve .....	22
2.3.4 e-arveldaja (RIK).....	22
2.3.5 Telema eInvoicing .....	23
3 Kasutatavad tehnoloogiad.....	24
3.1 Vue.js, Nuxt.js .....	24
3.2 Vuetify, Vue Material.....	25
3.3 npm .....	26
3.4 Actual Reports .....	27
4 E-arvete rakendus ja selle arenduskäik.....	28
4.1 Vue.js-i ja Nuxt.js-iga tutvumine .....	28
4.2 Vaated.....	28
4.2.1 Received Invoices.....	29
4.2.2 Received Invoices View .....	29
4.2.3 Upload .....	29
4.2.4 Created Invoices .....	30
4.2.5 Created Invoices Edit .....	30

4.2.6 Created Invoices View .....	30
4.2.7 New Invoice.....	30
4.2.8 Company Details .....	31
4.3 Suhtlus API-ga.....	31
4.4 Olulisemad loodud funktsionaalsused .....	31
4.4.1 Actual Reports-i kasutamine .....	32
4.4.2 Uute arvete loomine .....	33
4.4.3 Toodete salvestamine APIsse .....	36
4.4.4 XML faili sisu genereerimine.....	37
4.4.5 XML failide salvestamine API-sse.....	44
4.4.6 XML faili allalaadimine API-st.....	45
4.4.7 Arvete kustutamine.....	48
4.5 Loodud e-arve standardile vastavuse kontrollimine.....	49
5 Kokkuvõte .....	51
Kasutatud kirjandus .....	52
Lisa 1 – E-arvete rakenduse kasutusjuhud .....	54
Lisa 2 – Received Invoices vaade.....	55
Lisa 3 – Received Invoices View vaade.....	56
Lisa 4 – Upload vaade enne faili üleslaadimist .....	57
Lisa 5 – Upload vaade pärast faili üleslaadimist .....	58
Lisa 6 – Created Invoices vaade .....	59
Lisa 7 – Created Invoices Edit vaade .....	60
Lisa 8 – Created Invoices View vaade .....	61
Lisa 9 – New Invoice vaade .....	62
Lisa 10 – Company Details vaade .....	63
Lisa 11 – E-arvete API tabelid .....	64
Lisa 12 – Võimalikud päringud E-arvete rakenduse API tabelile .....	65
Lisa 13 – E-arvete rakenduse poolt genereeritud XML faili näidis .....	66

## Jooniste loetelu

Joonis 1. E-arve juurelemendi näidis.....	15
Joonis 2. Kohustuslikke osasid sisaldava <i>Header</i> elemendi näidis.....	15
Joonis 3. Element <i>Invoice</i> ja selle kohustuslikud osad.....	16
Joonis 4. Elemendi <i>InvoiceParties</i> kohustuslikud osad.....	16
Joonis 5. <i>SellerPartyRecord</i> tüübil põhineva <i>SellerParty</i> elemendi kohustuslikud osad. .....	17
Joonis 6. <i>BillPartyRecord</i> tüübil põhineva <i>BuyerParty</i> elemendi E-arvete rakenduses kasutatavad osad.....	17
Joonis 7. Elemendi <i>InvoiceInformation</i> E-arvete rakenduses kasutatavad osad. ....	17
Joonis 8. Elemendi <i>InvoiceSumGroup</i> E-arvete rakenduses kasutatavad osad. ....	18
Joonis 9. Elemendi <i>InvoiceItem</i> struktuur kohustuslike osadega. ....	18
Joonis 10. Elemendi <i>ItemEntry</i> struktuur E-arvete rakenduses. ....	19
Joonis 11. Elemendi <i>Addition</i> struktuur E-arvete rakenduses. ....	19
Joonis 12. Elemendi <i>ItemDetailInfo</i> struktuur E-arvete rakenduses. ....	19
Joonis 13. Elemendi <i>VAT</i> kohustuslikud osad.....	20
Joonis 14. Elemendi <i>PaymentInfo</i> näidis kohustuslike osadega.....	20
Joonis 15. <i>Footer</i> elemendi näidis.....	21
Joonis 16. Vuetify kaasamine projekti. ....	25
Joonis 17. Vue Material'i kaasamine projekti.....	26
Joonis 18. Material Design ikoonide lisamine projekti. ....	26
Joonis 19. Actual Reports'ile tehtav päring. ....	32
Joonis 20. Päring InvoicePdf komponendist Actual Reports'ile. ....	33
Joonis 21. InvoicePdf.vue HTML osa. ....	33
Joonis 22. Kõiki arvevormi välju sisaldav <i>computed property</i> . ....	33
Joonis 23. Näide arvevormi välja kirjeldusest.....	34
Joonis 24. Vue Material tüüpi väljade lisamine arvevormile. ....	34
Joonis 25. Arvevormi vasakpoolsed väljad. ....	34
Joonis 26. Arve salvestamine API-sse.....	35
Joonis 27. createJsonForAR meetod. ....	35



Joonis 28. Arvete tabelisse uue kirje lisamine.....	36
Joonis 29. uploadInvoiceProducts meetod. ....	36
Joonis 30. uploadProducts Vuex store action. ....	37
Joonis 31. Faili loomise kuupäeva genereerimine.....	37
Joonis 32. Arve tüübi määramine. ....	38
Joonis 33. Tasumisele kuuluva summa konverteerimine. ....	38
Joonis 34. E-arve JSON kujul. ....	40
Joonis 35. Toodete lisamine elementi <i>InvoiceItemGroup</i> . ....	41
Joonis 36. Elemendi <i>PaymentRefId</i> või <i>PaymentDescription</i> lisamine e-arve JSON objekti.....	42
Joonis 37. Maksekorralduse andmete lisamine e-arve JSON objekti.....	43
Joonis 38. Erimärkide kodeerimine ja JSON-i XML-iks konverteerimine. ....	43
Joonis 39. uploadInvoiceFile meetod. ....	44
Joonis 40. uploadFile Vuex store action. ....	44
Joonis 41. downloadXml meetod. ....	46
Joonis 42. generateBulkXml meetod.....	47
Joonis 43. download meetod. ....	48
Joonis 44. deleteInvoice meetod.....	48
Joonis 45. deleteRecord Vuex store action. ....	49
Joonis 46. DELETE meetod. ....	49
Joonis 47. deleteLocalRecord Vuex store mutatsioon. ....	49

## **Tabelite loetelu**

Tabel 1. Erimärkide kodeeringud. ....	14
---------------------------------------	----

# 1 Sissejuhatus

Igapäevaselt saadetakse suurtes kogustes arveid, millest Eestis on enamik paber- või PDF-kujul. Paber ning PDF-arvete sisestamine raamatupidamissüsteemidesse on aga väga aeganõudev ja veaohlik ning seda võib pidada isegi topelttöo tegemiseks. Lisaks on paberarvete väljatrükkimine ka ebamõistlik loodusressursside kasutus. Neid kõiki probleeme aitaks leevendada e-arvete kasutuselevõtt.

## 1.1 Taust ja probleem

Kahjuks Eestis e-arvete kasutamine veel suurt populaarsust kogunud ei ole, kuid alates 1. juulist 2019 on kogu Eesti avalik sektor e-arvetele üle minemas, mis tähendab, et riigiasutused enam paber- ega PDF-kujul arveid vastu ei võta. See on suur samm muudatuste suunas, sest ka erasektori ettevõtted peavad olema võimelised e-arveid koostama, et nad oleksid suutelised avaliku sektori asutustele arveid saatma. Ka erasektorile üldiselt oleks e-arvete kasutuselevõtt kasulik, kuna väheneks raamatupidajate töö, sest mitmekordse andmete ümbertrükkimise asemel tuleks vaid ühe korra andmeid sisestada.

Käesolev töö on valminud koostöös ettevõttega Erply. Erply on hulgi-, jae- ja teenindusettevõtetele suunatud veebipõhine majandustarkvara. Lisaks lao- ja kassatarkvarale pakub Erply ka veebirakenduste platvormi FNE.

Kuigi turul on juba mitmeid e-arvete teenusepakkujaid, kes pakuvad e-arvete operaatorteenust, ei ole Eesti turul veel ühtegi päris täpselt sellist lahendust, nagu seda on käesoleva lõputöö baasiks olev E-arvete rakendus. Erply E-arvete rakenduse eesmärgiks on pakkuda e-arvete keskkonda ka ilma operaatorita ning antud rakenduse kasutamine on tasuta.

## **1.2 Eesmärk**

Töö eesmärgiks on luua esimene versioon E-arvete rakendusest ettevõtte Erply sisendi põhjal. Arendatava rakendusega on võimalik Eesti e-arve standardile vastavaid e-arveid luua ning samas ka olemasolevaid, teistelt osapooltelt saabunud e-arve faile üleslaadida, et nende sisu PDF-vormingus vaadata. Selleks tuleb autoril tutvuda põhjalikult e-arvete ja nende kehtestatud nõuetega ning omandada autori jaoks uusi tehnoloogiaid, nagu näiteks Vue.js, Nuxt.js ja Actual Reports.

## **1.3 Ülevaade tööst**

Bakalaureusetöö teine peatükk räägib sellest, mis on e-arved ja kust saada infot ettevõtte e-arvete vastuvõtmise suutlikkuse kohta. Lisaks käsitleb peatükk Eesti e-arve standardi kohustuslikke ning teisi käesoleva lõputöö baasiks oleva E-arvete rakenduse (edaspidi töös viidatud kui E-arvete rakendus) osi ja kirjeldab, kes on e-arvete operaator ning millised e-arvete operaatorid Eestis eksisteerivad.

Kolmandas peatükis on lähemalt selgitatud E-arvete rakenduse arendamisel kasutatud tehnoloogiaid.

Neljas peatükk kirjeldab E-arvete rakenduse erinevaid vaateid, seda, kuidas suhtleb rakendus API-ga, toob välja mõned olulisemad loodud funktsionaalsused ning lõpuks räägib E-arvete rakenduses loodud XML failide valideerimisest.

Viies peatükk võtab kokku lõputöös saavutatud tulemused ja olulisemad järeldused.

## 2 E-arved

Tihti peale peavad inimesed ekslikult e-arvet PDF-arveks, kuid tegelikult on e-arve „...masinloetav arve, mis sisestatakse süsteemi üks kord ja milles olevad andmed on arvutite vahel masinloetavad [4].“

Vastavalt 20. veebruaril 2019 Riigikogus heaks kiidetud raamatupidamisseaduse seadusele on alates 1. juulist 2019 riigisektor täielikult üle minemas vaid e-arvetele, see tähendab, et alates juulikuust saab riigile saata arveid ainult e-arve kujul ja paber- ning PDF-arveid enam avalik sektor vastu ei võta. Muudatuse põhjuseks on tõsiasi, et paber- ja PDF-arvete töötlemine on ajakulukas ning vajab palju lisatööd. Tänu masinloetavatele e-arvetele saab riigi raamatupidamine minna kolme süsteemi asemel üle ühele arvete töötlemise süsteemile [4].

Eesti on e-arvete kasutamise poolest üks mahajääjatest. Eraettevõtete ja riigisektori vahel edastatakse kõigist arvetest vaid 19 protsenti e-arvetena. 64% edastatakse PDF-arvetena ja 17% paberarvetena [5].

Toomas Tõniste on öelnud, et muutes e-arvete saatmise avalikule sektorile kohustuslikuks, vähendab see arvete töötlemiseks kuluvat tööhulka ja -aega ning säästab maksumaksja raha, samuti ka tõstab raamatupidamisteenuse kvaliteeti [5].

### 2.1 E-arvete vastuvõtjate register

Riigikogu poolt 15. detsembril 2016. aastal vastu võetud raamatupidamise seaduse muutmise seaduse kohaselt peavad avaliku sektori ettevõtted olema võimelised e-arveid vastu võtma alates 2017. aasta 1. märtsist. Kui ettevõtte saadab riigisektori asutusele e-arve, peab see olema kooskõlas rahandusministri 11.04.2017 määruses nr 24 Masintöödeldava algdokumendi juhendi kehtestamine kehtestatud nõuetega või e-arveldamise Euroopa standardiga [4].

Selleks, et teada, millised ettevõtted e-arveid vastu võtavad, on loodud riiklik register „E-arvete vastuvõtjate register“. Sinna märgitakse kõik e-arveid vastu võtvad organisatsioonid, nii äri- ja mittetulundusühinguid kui ka sihtasutused ja riigi- ning kohaliku omavalitsuse asutused [4].

E-arvete vastuvõtjate register on osa Äriregistrist, seepärast on võimalik organisatsioonide e-arvete vastuvõtuvõimekuse kohta informatsiooni küsida Äriregistri teabesüsteemist sarnaselt organisatsiooni ülejäänud teabega [4].

## 2.2 Eesti e-arve standard

Ametlikult kehtiv Eesti e-arve standard on EVS 923:2014/AC:2017.

Järgnevalt on välja toodud olulisemad aspektid Eesti e-arve standardi rakendamiseks, toetudes rahandusministri 11.04.2017 määruse nr 24 „Masintöödeldava algdokumendi juhendi kehtestamine“ lisale „E-arvete juhend“ [1].

Eesti e-arvete XML failides kasutatakse UTF-8 kodeeringut ning XML failis erimärkide kasutamise korral tuleb need kodeerida vastavalt Tabelis 1 märgitule.

Tabel 1. Erimärkide kodeeringud.

&	&amp;
'	&apos;
>	&gt;
<	&lt;
„	&quot;

Eesti e-arve juurelemendiks on *E\_Invoice*, mis koosneb *Header*, *Invoice* ja *Footer* elementidest (struktuur nähtav Joonisel 1). *Header* määrab ära elemendid failipõhiselt, *Invoice* sisaldab infot ühe kindla arve kohta ja kordub iga järgmise arve puhul, *Footer* elemendis on kirjeldatud e-arve failis olevate arvete kogus ja nende samade arvete kogusumma.

```

<E_Invoice>
  <Header>
    ...
  </Header>
  <Invoice invoiceId="123456789" regNumber="1111111"
  channelId="EA" channelAddress="123123">
    ...
  </Invoice>
  <Invoice invoiceId="123456790" regNumber="2222222"
  channelId="EA" channelAddress="123123">
    ...
  </Invoice>
  <Footer>
    <TotalNumberInvoices>2</TotalNumberInvoices>
    <TotalAmount>451.25</TotalAmount>
  </Footer>
</E_Invoice>

```

Joonis 1. E-arve juurelemendi näidis.

Järgnevates peatükkides on täpsemalt välja toodud *Header*, *Invoice* ja *Footer* elementide kohustuslikud osad ning nende kirjeldused. Juhul, kui mõne ploki puhul on E-arvete rakenduse XML-is kasutatud ka mittekohustuslikke elemente, on need samuti täpsemalt lahti seletatud.

### 2.2.1 Header

*Header* (Joonis 2) koosneb failipõhistest elementidest ning selle kohustuslikud osad on *Date*, *FileId* ja *Version*.

```

<Header>
  <Date>2019-01-17</Date>
  <FileId>1122</FileId>
  <Version>1.2</Version>
</Header>

```

Joonis 2. Kohustuslikke osasid sisaldava *Header* elemendi näidis.

*Date* elemendiga tähistatakse e-arve loomise kuupäeva, *FileId* on unikaalne identifikaator, mis tagab, et sama identifikaatoriga dokumente ei töödeldaks mitmeid kordi ning *Version* märgib kasutuses oleva standardi versiooni.

## 2.2.2 Invoice

*Invoice* osa sisaldab ühe kindla arve informatsiooni ning neid elemente võib olla ühes failis ka mitu. *Invoice*'i kohustuslikeks elementideks on atribuudid *invoiceId*, *regNumber*, *sellerRegnumber* ning elementide plokid *InvoiceParties*, *InvoiceInformation*, *InvoiceSumGroup*, *InvoiceItem* ning *PaymentInfo*. Elemendi *Invoice* struktuuri on võimalik näha Joonisel 3.

*invoiceId* ehk arve unikaalne identifikaator garanteerib arve kordumatuse ühe faili siseselt, *regNumber* tähistab arve vastuvõtva poole registri- või isikukoodi ning *sellerRegnumber* müüja registrikoodi.

```
<Invoice invoiceId="A9200303961942" regNumber="10004252"
sellerRegnumber="11111111">
  <InvoiceParties>...</InvoiceParties>
  <InvoiceInformation>...</InvoiceInformation>
  <InvoiceSumGroup>...</InvoiceSumGroup>
  <InvoiceItem>...</InvoiceItem>
  <PaymentInfo>...</PaymentInfo>
</Invoice>
```

Joonis 3. Element *Invoice* ja selle kohustuslikud osad.

*InvoiceParties* elemendis on kirjeldatud arve osapoolte kohta käiv teave ning selle struktuuri kirjeldab Joonis 4. Elemendi nõutavateks osadeks on *SellerParty*, mis kujutab müüja andmeid ning mille struktuur vastab tüübile *SellerPartyRecord* ja *BuyerParty*, mis määratleb ostja informatsiooni ning vastab tüübile *BillPartyRecord*.

```
<InvoiceParties>
  <SellerParty>
    ...
  </SellerParty>
  <BuyerParty>
    ...
  </BuyerParty>
</InvoiceParties>
```

Joonis 4. Elemendi *InvoiceParties* kohustuslikud osad.

Elemendi *SellerParty* struktuuri kirjeldav *SellerPartyRecord* kajastab arvel esineva müüja informatsiooni ning koosneb kahest kohustuslikust osast *Name* ja *RegNumber*, mis



vastavalt tähistavad osapoolte nime ja registrikoodi (Joonis 5). *SellerParty* erineb muudest osapooltest registrikoodi kohustuslikkuse tõttu.

```
<SellerParty>
  <Name>Näidis OÜ</Name>
  <RegNumber>998877</RegNumber>
</SellerParty>
```

Joonis 5. *SellerPartyRecord* tüübil põhineva *SellerParty* elemendi kohustuslikud osad.

Tüüp *BillPartyRecord* sarnaneb struktuuri poolest suuresti *SellerPartyRecord* ile, defineerides arve osapoolte informatsiooni edasiandmiseks olulise teabe. Erinevus seisneb selles, et nõutav on vaid element *Name* ehk osapoolte nime sisaldav element, registrikoodi kajastamine kohustuslik ei ole, kuid E-arvete rakenduses on siiski ka elementi *RegNumber* kasutatud (Joonis 6).

```
<BuyerParty>
  <Name>Testfirma AS</Name>
  <RegNumber>123123</RegNumber>
</BuyerParty>
```

Joonis 6. *BillPartyRecord* tüübil põhineva *BuyerParty* elemendi E-arvete rakenduses kasutatavad osad.

*InvoiceInformation* plokk kajastab arve peamisi andmeid. Selle nõutavad osad on elemendid *Type* ehk arve tüüp atribuudiga *type* (atribuudi *type* väärtuseks on DEB ehk deebetarve või CRE ehk kreditarve); *DocumentName* ehk dokumendi nimi (E-arvete rakenduse puhul on 3 võimalikku varianti – *Invoice*, *Invoice-Waybill* või *Credit Invoice*); *InvoiceNumber* ehk arve number ning *InvoiceDate* ehk arve loomise kuupäev. Lisaks kohustuslikele elementidele kasutatakse E-arvete rakenduse puhul ka elementi *DueDate* ehk arve maksetähtpäeva. Elemendi struktuur koos kõigi E-arvete rakenduses kasutatavate osadega on nähtav Joonisel 7

```
<InvoiceInformation>
  <Type type="DEB"></Type>
  <DocumentName>Invoice</DocumentName>
  <InvoiceNumber>129592-2</InvoiceNumber>
  <InvoiceDate>2016-12-01</InvoiceDate>
  <DueDate>2016-12-15</DueDate>
</InvoiceInformation>
```

Joonis 7. Elemendi *InvoiceInformation* E-arvete rakenduses kasutatavad osad.

*InvoiceSumGroup* plokk on mõeldud arvega seotud summade esitamiseks (Joonis 8). Kohustuslikuks elemendiks on vaid *TotalSum* ehk arve maksumus kokku, kuid E-arvete rakenduses on kajastatud ka element *InvoiceSum* ehk arve summa ilma maksudeta.

```
<InvoiceSumGroup>
  <InvoiceSum>230.31</InvoiceSum>
  <TotalSum>255.51</TotalSum>
</InvoiceSumGroup>
```

Joonis 8. Elemendi *InvoiceSumGroup* E-arvete rakenduses kasutatavad osad.

*InvoiceItem* plokkis (Joonis 9) tuuakse välja arvel esinevate teenuste ja/või kaupade informatsioon. Standardikohaselt on selle nõutavaks elemendiks *InvoiceItemGroup* koos kohustusliku alamelemendiga *ItemEntry*. *InvoiceItemGroup* tähistab arve ridade gruppi ning on kohustuslik põhjusel, et iga arve peab omama sisu. *ItemEntry* kajastab ühte kindlat arverida. Nii *InvoiceItemGroup* kui ka *ItemEntry* elemente võib olla ühel arvel rohkem kui üks.

```
<InvoiceItem>
  <InvoiceItemGroup groupId="Clothes">
    <ItemEntry>
      ...
    </ItemEntry>
  </InvoiceItemGroup>
  <InvoiceItemGroup groupId="Footware">
    <ItemEntry>
      ...
    </ItemEntry>
    <ItemEntry>
      ...
    </ItemEntry>
  </InvoiceItemGroup>
</InvoiceItem>
```

Joonis 9. Elemendi *InvoiceItem* struktuur kohustuslike osadega.

Elemendi *ItemEntry* ainsaks nõutavaks osaks on *Description*, mis sisaldab kauba, teenuse või artikli nime või kirjeldust. Lisaks on E-arvete rakenduses kasutatud elemente *Addition*, *ItemDetailInfo*, *ItemSum*, *ItemTotal* ja *VAT*. Elemendi *ItemEntry* struktuuri iseloomustab Joonis 10.

```

<ItemEntry>
  <Description>Service</Description>
  <ItemDetailInfo>
    ...
  </ItemDetailInfo>
  <ItemSum>100.00</ItemSum>
  <Addition addCode="DSC">
    ...
  </Addition>
  <VAT>
    ...
  </VAT>
  <ItemTotal>120.00</ItemTotal>
</ItemEntry>

```

Joonis 10. Elemendi *ItemEntry* struktuur E-arvete rakenduses.

*Addition* element on mõeldud soodustuste ja juurdehindluste esitamiseks ning selle struktuur vastab tüübile *AdditionRecord*, mille kohustuslikud osad on atribuut *addCode* (väärtus DSC, kui tegemist on soodustusega, või CHR, kui tegemist on juurdehindlusega) ning element *AddContent* ehk juurdehindluse või soodustuse nimetus. E-arvete rakenduses on kasutatud ka elementi *AddRate*, mis tähistab juurdehindluse või soodustuse protsenti (Joonis 11).

```

<Addition addCode="DSC">
  <AddContent>Discount</AddContent>
  <AddRate>-20</AddRate>
</Addition>

```

Joonis 11. Elemendi *Addition* struktuur E-arvete rakenduses.

*ItemDetailInfo* väljendab kauba või teenuse üksikasjalikku teavet ning selle alamelementideks on *ItemUnit* ehk ühik, *ItemAmount* ehk kauba või teenuse hulk ning *ItemPrice* ehk kauba või teenuse ühe ühiku maksumus ilma maksudeta (Joonis 12).

```

<ItemDetailInfo>
  <ItemUnit>tk</ItemUnit>
  <ItemAmount>3</ItemAmount>
  <ItemPrice>110.78</ItemPrice>
</ItemDetailInfo>

```

Joonis 12. Elemendi *ItemDetailInfo* struktuur E-arvete rakenduses.

Element *VAT* edastab käibemaksu teavet. Selle struktuur vastab tüübile *VATRecord*, mille nõutavad osad on *VATRate* ehk käibemaksumäär ning *VATSum* ehk käibemaksu summa. Lisaks kasutatakse E-arvete rakenduses elementi *Currency*, mille all mõistetakse IOS 4217 standardile vastavat kolmetähelist valuuta koodi (Joonis 13).

```
<VAT>
  <VATRate>20</VATRate>
  <VATSum>22.47</VATSum>
  <Currency>EUR</Currency>
</VAT>
```

Joonis 13. Elemendi VAT kohustuslikud osad.

*ItemSum* elemendis kajastatakse kauba või teenuse rea lõppmaksumus ilma maksude ja soodustusteta, *ItemTotal* elemendis koos maksude ja soodustustega.

*PaymentInfo* plokk sisaldab arvest maksekorralduse loomise jaoks vajalikku informatsiooni (Joonis 14).

```
<PaymentInfo>
  <Currency>EUR</Currency>
  <PaymentDescription>Invoice number 321010
</PaymentDescription>
  <Payable>YES</Payable>
  <PayDueDate>2018-05-05</PayDueDate>
  <PaymentTotalSum>255.51</PaymentTotalSum>
  <PayerName>Maksja AS</PayerName>
  <PaymentId>321010</PaymentId>
  <PayToAccount>EE123456789123456789</PayToAccount>
  <PayToName>Test Ettevõte OÜ</PayToName>
</PaymentInfo>
```

Joonis 14. Elemendi *PaymentInfo* näidis kohustuslike osadega.

Selle kohustuslikud osad on *Currency*, *Payable*, *PaymentTotalSum*, *PayerName*, *PaymentId*, *PayToAccount*, *PayToName* ja valikuliselt kas *PaymentDescription* ehk maksekorralduse selgitus või *PaymentRefId* ehk arve viitenumber. *Currency* tähistab kolmetähelist ISO standardile vastavat valuuta koodi; *Payable* elemendist sõltub, kas arve kuulub tasumisele (väärtuseks on YES) või mitte (väärtuseks on NO), kusjuures kui arve tuleb tasuda, on kohustuslik ka element *PayDueDate* ehk arve maksetähtaeg; *PaymentTotalSum* märgib tasumisele kuuluvat summat, *PayerName* ostja nime, *PaymentId* arve numbrit, *PayToAccount* müüja makse laekumiskontot ning *PayToName* müüja nime.

### 2.2.3 Footer

*Footer* element annab edasi failis esinevate arvete arvu ja nende kogusumma ning selle struktuur on esitatud Joonisel 15.

```
<Footer>
  <TotalNumberInvoices>1</TotalNumberInvoices>
  <TotalAmount>125.64</TotalAmount>
</Footer>
```

Joonis 15. *Footer* elemendi näidis.

Nõutavad alamelemendid on *TotalNumberInvoices*, mis näitab *Invoice* elementide ehk arvete arvu failis ja *TotalAmount*, mis kajastab *PaymentTotalSum* elementide kogusummat, ilma et arvestataks erinevusi valuutade vahel. Mõlemad alamelemendid on vajalikud kontrollmehhanismiks, et olla kindel faili kooskõllalisuses.

## 2.3 E-arvete operaatorid Eestis

Selleks, et e-arveid lähetada, on vaja sõlmida leping e-arvete operaatoriga, pärast mida saab ettevõtte aktiveerimisevõtme, et oma tarkvara operaatoriga liidestada. Seejärel saavad e-arved liikuma hakata [6].

E-arvete operaatorid sarnanevad suuresti telekomi firmadele, kuid kõnede asemel liigutatakse arveid [7]. „Operaator on kui keskjaam, kuhu kõik arved sisse tulevad, ja sealt suunatakse need n.ö õigele aadressile [8].“ Operaatorite ülesanne on kanda hoolt, et arve jõuaks kohale, informatsioon on kaitstud ning keskkond on kasutajale piisavalt arusaadav [7].

Erinevatel e-arvete operaatoritel on võimekus saata arveid edasi ka teistele e-arvete operaatoritele.

Eestis pakuvad e-arvete operaatori teenust Amphora, Fitek, Omniva, Registrate ja Infosüsteemide Keskus (RIK) ja Telema.

### 2.3.1 Arved.ee, Fitek Premium (Fitek)

Arved.ee on Fiteki halduses olev e-arvete keskkond, mille peamiseks eesmärgiks on e-arvete populariseerimine ning e-arvete osatähtsuse tõstmine Eestis. Antud keskkond annab võimaluse nii era- kui ka juriidilistel isikutel esitada e-arve soove, uurida jões olevaid e-arvete soove keskkonnaga liitunud ettevõtete osas ning anda sisse e-arvete koondtellimusi. Samuti on Arved.ee keskkonnas võimalus firmadel e-arveid koostada ja oma klientidele erinevatesse Eestis kasutatavatesse kanalitesse edasi saata [9].

Lisaks Arved.ee keskkonnale on Fiteki hallata ka Fitek Premium, mis kujutab endast täisteenuste paketti ostuarvete vastuvõtmiseks ning on loodud ettevõtetevahelise äritegevuse tulemuslikuks organiseerimiseks. Fitek Premium on mõeldud firmadele, kellel on tarvidus töödelda suurt kogust ostuarveid [10].

### **2.3.2 Arvekeskus (Omniva)**

Arvekeskus on Omniva e-arvete keskkond, mis lisaks põhilisele e-arvete loomise ning nende pakkuja ja ostja vahel liigutamise funktsioonile pakub võimalust ostuarveid meilis või nutitelefonis kinnitada, paber- ja PDF-arveid digiteerida ning arveid arhiveerida [8].

Arveid on võimalik koostada nii Arvekeskuses kui ka oma raamatupidamissüsteemis ning Arvekeskus on liidestatud kõikide populaarsemate tarkvaradega. Lisaks Eesti e-arvete vormingute toetamisele toetatakse lähitulevikus ka Euroopa Liidu formaati ning tehakse koostööd Pageroga, et üle Euroopa laiemat võrgustikku võimaldada. Samuti on Arvekeskus liitunud PEPPOLiga, tänu millele on peagi võimalus Arvekeskuse kaudu saata arveid ka kõigile PEPPOLiga ühinenud ettevõtetele Euroopas [11].

### **2.3.3 Amphora e-arve**

Amphora e-arve on e-arvete operaator ja halduskeskkond, mille kasutamine sisaldub Amphora kuumaksumuses – see tähendab, et Amphora kliendid Amphora e-arvete kasutamise eest lisatasu maksma ei pea [12].

Peale selle võimaldab Amphora tasulist PDF-arvete digiteerimise teenust kahe teenuspakkuja, Envoice ja Omniva, kaudu. Nii müügi- kui ka ostuarveid võetakse vastu kõikidelt Eestis olevatelt operaatoritelt ja DVK ehk Dokumendivahetuskeskuse ning meili kaudu [12].

### **2.3.4 e-arveldaja (RIK)**

e-arveldaja on veebipõhine raamatupidamistarkvara, mis asub e-äriregistri ettevõtjaportaalis. Tarkvara võivad kasutada nii ettevõtjad ise kui ka nende poolt volitatud isikud, näiteks raamatupidaja [13].

e-arveldajas on võimalik saata riigisektori ettevõtetele e-arveid ilma tasuta, koguseliselt piiramatult ning tähtajata. Peale selle pakub e-arveldaja ka majandustarkvaradele e-arve operaatori teenust [13].

e-arveldaja on ühendatud Eestis olemasolevate operaatoritega ja e-arveid on võimalik edastada kõikidele, sõltumata sellest, milline on vastuvõtva organisatsiooni e-arvete operaator [4].

### **2.3.5 Telema eInvoicing**

Telema eInvoicing on Telema pakutav lahendus e-arvete haldamiseks. Et Telema eInvoicing'ut kasutada, tuleb e-arve koostada oma majandustarkvaras ning arve väljasaatmisel teisendab Telema e-arve sobivasse vormingusse ja saadab selle edasi määratud saajale. Telemaga on võimalik saata e-arveid kõikidele, sõltumata sellest, millise operaatoriga nad liitunud on [14].

## 3 Kasutatavad tehnoloogiad

Käesolev peatükk annab lühiülevaate tähtsamatest E-arvete rakenduses kasutatavatest tehnoloogiatest ning selgitab, mille jaoks on üht või teist tehnoloogiat kasutatud.

### 3.1 Vue.js, Nuxt.js

Vue.js (lühidalt Vue) on progressiivne JavaScripti raamistik, mis on mõeldud kasutajaliideste ja *single-page-application*'ite ehk SPA-de loomiseks. Erinevalt teistest terviklikest raamistikest disainiti Vue kohe alguses olema järkjärgult kasutuselevõetav. Tuumik *library* keskendub vaid *view layer*'ile ning seda on lihtne teiste teekide ja olemasolevate projektidega integreerida [15].

Nuxt.js on raamistik Vue.js rakenduste loomiseks. Valida on võimalik *Universal*, *Static Generated* ja *Single Page* rakenduste vahel. Nuxti peamiseks käsitlusalaks on kasutajaliidese renderdamine, mis samal ajal aitab ka üldistada kliendi-serveri jaotust [16], tehes seda kontseptuaalselt lihtsamaks.

Nuxt.js-i eesmärgiks on pakkuda nii paindlikku raamistikku, et seda oleks võimalik kasutada projekti peamise põhjana või olemasoleva Node.js põhjal tehtud projekti lisana. Nuxt.js eelseadistab kogu serveri poolt renderdatavate Vue.js rakenduste jaoks vajaliku konfiguratsiooni, et arendamine oleks mugavam [16].

Nuxt.js-il on mitmeid lisasid, mis aitavad kliendi- ja serveripoolsel arendamisel, näiteks *Asynchronous Data*, *Middleware* ja *Layouts*.

Nuxt.js koosneb järgnevatest osadest:

- Vue 2
- Vue Router
- Vuex (kaasatud ainult siis, kui kasutatakse Vue *store*'i)
- Vue Server Renderer (välja arvatud, kui rakenduse tüübiks on SPA)
- vue-meta

Lisaks kasutab Nuxt.js webpack-i koos vue-loader-iga ning babel-loader-it, et koodi kokku pakkida, osadeks jagada ja minimeerida [16].



Nuxt.js funktsionaalsused hõlmavad endas Vue failide kirjutamist (\*.vue); automaatset koodi tükeldamist; serveripoolset renderdamist; võimsat asünkroonsete andmete marsruutimise süsteemi; staatiliste failide teenindamist; ES2015+ transpileerimist; JavaScripti ja CSS keele kokku pakkimist ja minimeerimist; *<head>* elemendi haldamist; arenduskeskkonnas *Hot Module Replacement*'i ehk rakenduse töötamise ajal moodulite vahetamist, lisamist või eemaldamist ilma täieliku taaskompileerimiseta [17]; Sass, Less, Stylus eelprotsessoreid; võimekust HTTP/2 *push header*'ite kasutamiseks ning modulaarse arhitektuuri abil laiendamist [16].

E-arvete rakenduse projekti loomiseks kasutas autor Nuxt.js tööriista `create-nuxt-app`, mis genereerib projekti struktuuri (baaskaustad ja -failid), mis teeb arendamisega alustamise väga lihtsaks ja kiireks.

Rakenduse konfigureerimine toimub failis `nuxt.config.js`, kus muu hulgas saab projekti lisada globaalselt kasutatavaid CSS faile ja väliseid CSS-i *library*'eid, määratleda arendus- ja tootmiskeskonna erisusi ning kaasata rakenduses kasutatavaid pluginaid.

### 3.2 Vuetify, Vue Material

Vuetify on pidevalt edasi arendatav veebiarenduse jaoks mõeldud Material Design komponentide raamistik. Vuetify toetab serveripoolset renderdamist (SSR ehk *server-side rendering*), SPA-sid ehk *Single Page Application*'eid, progressiivseid veebirakendusi (PWA ehk *Progressive Web Application*) ja tavapäraseid HTML lehti [18].

Selleks, et Vuetify-d Nuxt.js projekti kaasata, tuleb see lisada *dependency*'ina `package.json` faili, laadida alla selle npm *package* ning lisada Vuetify `nuxt.config.js` failis moodulite alla (Joonis 16).

```
modules: [  
  '@nuxtjs/vuetify'  
]
```

Joonis 16. Vuetify kaasamine projekti.

E-arvete rakenduses on Vuetify komponentidest kasutatud *Data table* ehk *v-data-table* komponenti arve andmete tabelis näitamiseks ning *Snackbar*'i erinevate teavituste kuvamiseks. *Data table* komponent võimaldab sorteerida tabelis olevaid andmeid,

otsinguväljale kirjutades leida sobivaid vasteid, tabeli kirjeid lehekülgedeks küljendada ja kuvada soovitud arv vasteid ühel lehel.

Lisaks Vuetify-le on käesolevas lõputöös kasutatud ka Vue Material-i, mis kujutab endast samuti Material Design komponentide raamistikku.

Vue Material komponentide kasutamiseks tuleb need npm-i või Yarn-iga alla laadida ning projekti importida. Võimalik on nii üksikuid komponente importida kui ka importida kõikide komponentide kogum [19]. E-arvete rakenduse puhul on imporditud kõik Vue Material-i komponendid *nuxt.config.js* failis moodulina (Joonis 17).

```
modules: [  
  'nuxt-vue-material'  
]
```

Joonis 17. Vue Material'i kaasamine projekti.

Material Design ikoonide kasutamiseks on vastav teek lisatud *nuxt.config.js* failis *header*'isse *link*'ina (Joonis 18).

```
head: {  
  link: [  
    {  
      rel: "stylesheet",  
      href:  
        "https://fonts.googleapis.com/icon?family=Material+Icons"  
    }  
  ]  
}
```

Joonis 18. Material Design ikoonide lisamine projekti.

Vue Material-i komponentidest on rakenduses kasutatud nuppe (*md-button*) ja erinevaid väljasid (*md-field*). Väljatüüpide kasutamise lihtsustamiseks ning ühtse stiili hoidmiseks on Erply arendajad loonud Vue Material-il põhineva *library* nimega FieldFormElement. *Library* kasutamiseks tuleb see *dependency*'na lisada *package.json* faili, npm-i või Yarn-i abil alla laadida ning seejärel sobivasse komponenti importida.

### 3.3 npm

npm-i võib seletada mitmeti. npm-iks kutsutakse Node.js pakettide haldurit, mis loodi avatud lähtekoodiga projektina, et aidata JavaScripti arendajatel oma paketeeritud

koodimooduleid jagada [20]. Samas on npm ka maailma suurim tarkvara register, kuhu kõik avatud lähtekoodiga tegeleda soovivad arendajad saavad panustada ning kasutada teiste arendajate poolt loodud koodi [21].

npm koosneb kolmest eraldiseisvast osast: veebilehest, CLI-st ehk *Command Line Interface*'ist ja registrist. Veebilehe abil on võimalik leida soovitud koodipakette, CLI on mõeldud terminali aknas npm-i kasutamiseks ja register kujutab endast suurt avalikku JavaScripti tarkvara andmebaasi ja seda ümbritsevaid metaandmeid [21].

Käesolevas töös on autor kasutanud näiteks npm *library*'t xml-js [22], et konverteerida e-arve andmed JSON kujult XML kujule (kirjeldatud täpsemalt peatükis 4.4.4).

### **3.4 Actual Reports**

Actual Reports on pilvepõhine dokumendiloome platvorm, mis aitab lihtsustada dokumentide koostamise protsessi [23].

Käesolevas lõputöös on kasutatud Actual Reports-i toodet PDF Generator API, mis on mõeldud arendajatele. Tootega kaasneb brauseripõhine dokumentide redaktor, millega on võimalik luua ja hallata erinevaid dokumendimalle ning oma tarkvarast API-sse andmeid saates loodud dokumendimallide põhjal PDF ja HTML dokumente luua. PDF Generator API kasutamiseks tuleb see kõigepealt oma rakenduse külge integreerida [23].

Antud funktsionaalsuse kasutamise jaoks on ettevõtte Erply arendaja poolt loodud rakendus Ui Actualreports, mille külge on PDF Generator API integreeritud. Põhjalikumalt on Actual Reports-i toimimist E-arvete rakenduses käsitletud peatükis 4.4.1.

## 4 E-arvete rakendus ja selle arenduskäik

Antud peatükk annab ülevaate E-arvete rakenduse toimimisest, kasutusjuhtudest ning tähtsamatest loodud funktsionaalsustest. Tegemist on rakenduse esimese versiooniga, mis on inglisekeelne, mõeldud kasutamiseks laua- ja sülearvutites ning lõputöö koostamise hetkel rakendus ühegi e-arve operaatoriga ühendatud ei ole.

Sisend antud *app*'i tegemiseks tuli ettevõttelt Erply – loodud oli kokkuvõtlik nõuete dokumentatsioon, mis kirjeldas ära vajalikud funktsionaalsused, ning Erply UX disainer koostas *mock*'i, et oleks rakenduse kujunduse puhul millestki juhinduda. Enne *app*'i kallal töötama asumist tuli põhjalikult analüüsida püstitatud nõudeid, mõelda, milliseid tehnoloogiaid kasutada, kuid suures osas sai eeskuju võtta teistest juba loodud Erply rakendustest.

*App*'i loomisel on kasutatud ka mõningaid teiste Erply arendajate poolt loodud funktsionaalsusi, näiteks *menu-bar* komponenti ja varem käesolevas töös mainitud Actual Reports-iga integreeritud rakendust.

### 4.1 Vue.js-i ja Nuxt.js-iga tutvumine

Autori kogemus Vue-ga enne E-arvete rakenduse arendamist oli praktiliselt olematu – ülikooliõpingute ajal kokkupuuteid Vue-ga ei esinenud ning selle omandamist alustas autor alles tööl käies, seega oli antud ülesanne heaks väljakutseks. Esialgu vaatas autor Pluralsight [24] lehel erinevaid õpetusvideoid, hiljem olid peamisteks infoallikateks Nuxt.js-i ja Vue veebilehel olevad raamistike dokumentatsioonid.

### 4.2 Vaated

Rakenduse arendamist alustas autor erinevate vaadete loomisest, mida on kokku 8. Vaadete failid asuvad *pages* kaustas, mille põhjal Nuxt.js automaatselt genereerib *vue-router* konfiguratsiooni. Järgnevates peatükkides on selgitatud, mille jaoks erinevaid vaateid kasutatakse ning samuti on lisatud pildid nende vaadete illustreerimiseks. Lisas 1 asub rakenduse kasutusjuhtude diagramm, mis näitab ära, milliseid tegevusi rakenduses teha on võimalik.

### 4.2.1 Received Invoices

Received Invoices vaates on võimalik näha rakendusse üleslaetud arveid tabeli kujul (Lisa 2). Tabelis kuvatakse arve loomise kuupäev, arvetüüp (*Invoice*, *Invoice-Waybill* või *Credit Invoice*), arvenumber, ettevõtte nimi, kellelt on arve saadud, ning mis on arve kogumaksumus. Tabelireale vajutades avatakse uus vaade (täpsem kirjeldus peatükis 4.2.2), kus on võimalik näha arvet PDF-kujul.

Tabeli kirjeid saab järjestada vajutades tulpade pealkirjadele ning lisaks on võimalik kasutada otsingut. Otsinguväljale kirjutatu põhjal filtreeritakse välja ainult need vasted, mis sisaldavad antud sõna või sõnaosa.

Märkides linnukesega ära ühe arverea ja vajutades nuppu „Download XML“, laetakse alla XML fail, mis sisaldab vaid selle konkreetse arve andmeid. Received Invoices tabel on vajalik selleks, et hoida kõiki saabunud e-arveid ühes kohas ning kui arve on mujalt ära kustutatud, on ta rakenduses endiselt alles. Kui aga kasutaja märgib ära mitu arverida ja vajutab „Download XML“ nupule, moodustatakse *bulk* XML fail, kus on kokku pandud mitu arvet (arve XML koosneb mitmest *Invoice* elemendist).

### 4.2.2 Received Invoices View

Received Invoices View vaade kuvab saabunud arvete tabeli kõrval ühe konkreetse, avatud arve PDF-kujul (Lisa 3). PDF-faili genereerimiseks on kasutatud Actual Reports-i ning selle kasutamise tehnilisem kirjeldus asub peatükis 4.3.2.

### 4.2.3 Upload

Selleks, et rakendusse e-arveid üles laadida, on loodud vaade Upload (Lisa 4, Lisa 5), mis avaneb, kui Received Invoices lehel vajutada „+“ nupule. Arveid on võimalik üles laadida nii *drag and drop* abil kui ka vastavale väljale klikkida ning sobivast kaustast fail valida, misjärel kuvatakse arve PDF-kujul Actual Reports-i abil. Arve on võimalik PDF-kujul alla laadida, vajutades PDF-i eelvaate aknas *Download* nupule, või üleslaetud arve Received Invoices tabelisse importida, klikkides „Import“ nupule.

#### 4.2.4 Created Invoices

Created Invoices vaade sarnaneb suuresti Received Invoices vaatele (Lisa 6). Vahe on selles, et Created Invoices lehel olevas tabelis kuvatakse vaid rakenduses endas loodud e-arded ning juba loodud e-ardeid on võimalik muuta, kui need ei ole veel välja saadetud. Tabelis kuvatakse arve loomise aeg, arvetüüp, ostja nimi, arve number, arve kogusumma ning staatus („*Saved*“ või „*Sent*“).

Loodud arveid on võimalik ka kustutada ning loodud arve XML faili alla laadida. Selleks tuleb sobiv arve või arved linnukesega märgistada ja vajutada *save\_alt* ikoonile. Samuti on rakenduse järgmises versioonis võimalik arveid *app*'ist edasi saata näiteks raamatupidajale, kuid see funktsionaalsus hetkel implementeeritud ei ole – „*Send*“ nupule vajutamine muudab lihtsalt arve staatus „*Saved*“ olekust olekusse „*Sent*“, et näidata, milline peaks tulevikus „*Sent*“ staatusena arve vaade välja nägema.

Lisaks on selles vaates olemas ka otsingufunktsioon, mis toimib samamoodi nagu Received Invoices vaate puhul – kuvatakse kirjed, mis kas täielikult või osaliselt vastavad otsingusõnale ning kirjeid on tulba pealkirjadele vajutades võimalik sorteerida.

#### 4.2.5 Created Invoices Edit

Created Invoices Edit vaade (Lisa 7) avaneb, kui vajutada Created Invoices tabelis „*Saved*“ staatusena arvele. Selles vaates kuvatakse arvevorm avatud arve andmetega täidetult ning kõiki sisestatud andmeid on võimalik muuta ja arvele uusi tooteid lisada.

#### 4.2.6 Created Invoices View

Created Invoices View vaade (Lisa 8) avatakse Created Invoices tabelis „*Sent*“ staatusena arvele vajutades. Antud vaates kuvatakse üks konkreetne rakenduses loodud e-arve PDF-kujul ning selle arve andmeid enam muuta pole võimalik.

#### 4.2.7 New Invoice

New Invoice vaade (Lisa 9) on mõeldud selleks, et E-ardete rakenduses uusi e-ardeid luua. Kohustuslik on täita ära *Invoice type*, *Company name*, *Registration number*, IBAN, *Invoice no.*, *Date*, *Due date* ja *Currency* väljad. Kuna makse tegemise puhul on kohustuslik täita kas viitenumbri või selgituse väli, siis juhul kui *Reference number* väli

on täidetud, lisatakse e-arve XML-i element *PaymentRefId*. Kui Reference number väli on tühi, lisatakse XML-i element *PaymentDescription* ehk maksekirjeldus kujul ``${ arvetüüp } number ${arve number}``, näiteks „*Invoice number 111*“.

#### 4.2.8 Company Details

Company Details lehel (Lisa 10) saab täita vajalikud väljad e-arve maksekorralduse osa jaoks. Sellel lehel tuleb täita *Company name*, IBAN, *Registration number* ja *Currency* väljad, et arvete loomisel oleks võimalik neid andmeid kasutada.

### 4.3 Suhtlus API-ga

Selleks, et näiteks loodud arveid, nendega seotud tooteid ja faile säilitada ning neid tabelis kuvada, tuleb need ka kuskile salvestada. Selle jaoks on Erply *app*'ide keskkonnas E-arvete rakendusega ühendatud rakendus E-Invoicing API, mis kujutab endast üldistatult andmebaasi rakenduse teabe salvestamiseks ning seal on ära kirjeldatud kõik rakenduses eksisteerivad tabelid ning nende väljad ja väljade tüübid. Näide ühest rakenduse API tabelist asub Lisas 11 ning Lisa 12 illustreerib kõiki võimalikke API tabelitele tehtavaid päringuid `CREATED_INVOICES` tabeli näitel.

API-le tehtavate päringute loogika jaoks on loodud rakendusse *Vuex store* `ErplyApp.js`, kus *actions* ploki all on ära kirjeldatud erinevad funktsioonid, mida *app*'i vaadetes on võimalik API päringute tegemiseks välja kutsuda.

Päringute tegemise mugavamiseks on Erply arendajate poolt loodud *library* `ErplyCloudApi`, kus on kirjeldatud ära kõik erinevad HTTP päringud, mida teha on võimalik. Päringute tegemisel kasutatakse *axios*'t. Näiteid *app*'is tehtavates API päringutest leiab järgmise peatüki alampeatükkidest.

### 4.4 Olulisemad loodud funktsionaalsused

Käesolev peatükk annab ülevaate mõningatest tähtsamatest E-arvete rakenduse funktsionaalsustest.

#### 4.4.1 Actual Reports-i kasutamine

Actual Reports-i kasutamiseks tehakse E-arvete rakendusest päring Actual Reports-iga integreeritud Erply *app*'ide *sandbox* keskkonna rakendusele nimega UI Actualreports. See rakendus on *dependency*'ina E-arvete rakenduse külge *include*'itud.

Päringute tegemiseks on loodud Vuex *store* documentGeneration.js, mis sisaldab endas *action*'it loadTemplateFor (Joonis 19), mis annab päringuga kaasa dokumendi andmed JSON stringina, malli ID, dokumendi formaadi ning soovitud väljundi. Kaasa pandav malli ID tähistab *sandbox* keskkonna rakenduses Demo Ui Actualreports E-arvete rakenduse jaoks loodud dokumendimalli numbrit.

```
loadTemplateFor({commit}, document) {
  commit('beginLoading')
  this.$axios.post(process.env.apiUrl+"/service/ui-
  actualreports/templates/" +
  process.env.arTemplateId + "/generate",
  document,
  {params:
  {
    format: 'pdf',
    output: 'base64'
  }
  }).then(res => {
    commit('setDocumentUrl', res.data.url)
    commit('setDocumentBase64',
    res.data.document)
  })
}
```

Joonis 19. Actual Reports'ile tehtav päring.

Actual Reports-i funktsionaalsust on kasutatud näiteks InvoicePdf.vue komponendis, mida sisaldavad vaated Received Invoices View ja Created Invoices View.

InvoicePdf komponendist tehakse päring Actual Reports-ile, kui arve andmed muutuvad, mille kindlaks tegemiseks on kasutatud *watch*'i (Joonis 20). loadTemplateFor *action*'ile JSON stringina kaasa antavad andmed on eelnevalt salvestatud arvete tabelis arve kirje väljale *JsonForAr*.



```

watch: {
  invoice() {
    this.loadTemplateFor({...this.invoice.jsonForAr})
  }
}

```

Joonis 20. Päring InvoicePdf komponendist Actual Reports'ile.

InvoicePdf komponendi HTML osa on väga lühike (Joonis 21) ja sisaldab elementi *object*, millele antakse sisse Actual Reports-ist tulnud Base64 kujul vastus arve PDF-ina kuvamiseks.

```

<template>
  <div class="col-lg-6 pr-lg-0 pl-2 column">
    <div class="page">
      <object v-if="!loading"
        :data="'data:application/pdf;base64,' + documentBase64"
        class="invoice-preview"/>
    </div>
  </div>
</template>

```

Joonis 21. InvoicePdf.vue HTML osa.

Tulemuseks on PDF-kujul kuvatav arve, mis teeb e-arve andmed oluliselt arusaadavamaks ja loetavamaks.

#### 4.4.2 Uute arvete loomine

Uute arvete loomine toimub New Invoice vaate alamkomponendis InvoiceForm.vue. InvoiceForm komponendi HTML osa sisaldab kahte tüüpi väljasid – Vue Material tüüpi väljad, mille jaoks on kasutatud Erply arendajate poolt loodud *library*'it FieldFormElement, ning tavalised HTML *input* väljad. Kõik väljad on ära kirjeldatud *schemas* kaustas olevas failis invoice.js, mis imporditakse InvoiceForm komponenti ning milles sisalduv info omistatakse *fields computed property*'le (Joonis 22).

```

fields() {
  return schemas.invoice.editFields
},

```

Joonis 22. Kõiki arvevormi välju sisaldav *computed property*.

Joonisel 23 on näidised välja toodud *Customer name* välja kirjeldus.

```

customerName: {
  name: "customerName",
  type: "text",
  size: 254,
  required: true,
  error: false
},

```

Joonis 23. Näide arvevormi välja kirjeldusest.

Vue Material tüüpi väljade lisamist iseloomustab Joonis 24, kus iga *computed property* leftFields (Joonis 25) väärtuse ehk arvevormil vasakul pool paikneva välja kohta luuakse üks FieldFormElement element.

```

<div v-for="field in leftFields" :key="field.name">
  <FieldFormElement
    :record="invoice"
    :field="field"
    :label="$t('fields.'+ field.name)"
  />
</div>

```

Joonis 24. Vue Material tüüpi väljade lisamine arvevormile.

```

leftFields() {
  const {
    invoiceType,
    customerName,
    registrationNo,
    iban,
    referenceNo
  } = this.fields

  return { invoiceType, customerName, registrationNo,
    iban, referenceNo }
},

```

Joonis 25. Arvevormi vasakpoolsed väljad.

Arve salvestamiseks tuleb vajutada „Save“ nupule, misjärel käivitatakse save meetod (nähtav Joonisel 26).

```

if (isUnique && companyDetailsInserted && !hasErrors &&
hasAtLeastOneProduct) {
  this.createJsonForAR(productsArray)
  this.invoice.status = SAVED
  if (this.$route.name === 'new') {
    this.createRecord({
      record: this.invoice,
      callback: this.saveCallback,
      url: 'createdInvoices'
    })
  } else {
    this.updateRecord({
      record: this.invoice,
      callback: this.saveCallback,
      url: 'createdInvoices'
    })
  }
}
}

```

Joonis 26. Arve salvestamine API-sse.

Kui sisestatud arvenumber on unikaalne, Company Details vaates on kõik andmed sisestatud, arvevormi väljadele andmete sisestamisel ei ole eksitud valideerimise reeglite vastu ja arvele on lisatud vähemalt üks toode, käivitatakse createJsonForAR meetod (Joonis 27), mis loob Actual Reports-i jaoks vajaliku JSON stringi kõigi arve ja sellele lisatud toodete andmetega, ning arvele omistatakse staatus „*Saved*“.

```

createJsonForAR(productsArray) {
  if (this.$route.name === 'created-id-edit') {
    delete this.invoice['jsonForAr']
  }
  productsArray.map(product => {
    if (!product.quantity) product.quantity = 1
    if (!product.unit) product.unit = '-'
    if (!product.discount) product.discount = 0
    if (!product.vat) product.vat = 0
    return product
  })
  const actualReportsData = Object.assign(
  { products: productsArray }, this.invoice)
  this.invoice['jsonForAr'] = actualReportsData
}

```

Joonis 27. createJsonForAR meetod.

Seejärel läheb käima `createRecord` päring (Joonis 28), millele antakse parameetrina *record* kaasa arve objekt *invoice*, kuhu on salvestatud kõik arvevormi väljadele sisestatud andmed, *callback* ehk tegevused, mis tuleks teha vahetult pärast arvete tabelisse uue kirje loomist, ning kuna arvete tabeleid on rohkem kui üks (`RECEIVED_INVOICES`, `CREATED_INVOICES`), antakse parameetrina kaasa ka *url*, et täpsustada, millisele tabelile päring teha tuleb.

```
createRecord({ state, dispatch }, { record, callback, url }) {
  const InvoicesApi = new ErplyCloudApi(process.env.apiUrl[url])
  state.loading = true
  InvoicesApi.CREATE(record, () => {
    dispatch("fetchRecords", { callback, url })
  })
}
```

Joonis 28. Arvete tabelisse uue kirje lisamine.

Kuna `save` meetodit kasutab ka `Created Invoices Edit` vaade, mis on mõeldud olemasolevate arvete muutmiseks, on `save` ja `createJsonForAR` meetodid loodud *mixin*'itena ning sisaldavad mõningaid tingimuslauseid, et sobida mõlema vaate jaoks.

#### 4.4.3 Toodete salvestamine APIsse

`createRecord` päringu üheks parameetrik on *callback* väärtusega `saveCallback`, mis muuhulgas kutsub välja meetodi `uploadInvoiceProducts` (Joonis 29).

```
uploadInvoiceProducts() {
  let savedInvoice = this.getSavedInvoice()
  if (savedInvoice) {
    this.products.map(product => {
      product.invoice = savedInvoice.id
      return product
    })
    this.uploadProducts({ products: this.products,
      callback: null })
  } else {
    console.log('no savedInvoice')
  }
}
```

Joonis 29. `uploadInvoiceProducts` meetod.

Tooted salvestatakse toodete tabelisse (PRODUCTS) ning vahetult enne uploadProducts päringu tegemist (Joonis 30) omistatakse iga toote külge arvenumber välisvõtmena, et salvestatud tooteid oleks võimalik arvega seostada.

```
uploadProducts({ }, { products, callback }) {
  ProductApi.CREATEBULK(products, () => {
    this.loading = false
    if (callback) callback()
  })
}
```

Joonis 30. uploadProducts Vuex store action.

Kui Created Invoices Edit vaates on vaja arve andmeid muuta, lisatakse arvevormi need tooted, mis on valitud arvega seotud välisvõtme ehk arve ID kaudu.

#### 4.4.4 XML faili sisu genereerimine

E-arve XML faili genereerimine koosneb kahest osast – esiteks tuleb koostada faili sisu ning seejärel fail ise luua. Käesolevas peatükis on käsitletud sisu loomise osa ning peatükk 4.4.5 räägib täpsemalt faili loomise ning API-sse salvestamise protsessist.

Kuna XML faili genereerimine toimub kahes vaates (New Invoice ja Created Invoices Edit), oli koodi dubleerimise vältimiseks mõistlik luua XML-i genereerimise funktsionaalsuse jaoks *mixin* ning see seejärel sobivatesse komponentidesse importida.

Üheks *Header* elemendi kohustuslikuks osaks on *Date* element ehk arve faili loomise kuupäev, mille genereerimist on kirjeldatud Joonisel 31.

```
let invoiceDate
const date = new Date()
const year = date.getFullYear()
let month =
date.getMonth() + 1 > 9 ? date.getMonth() + 1 :
`0${date.getMonth() + 1}`
const day = date.getDate() > 9 ? date.getDate() :
`0${date.getDate()}`
invoiceDate = `${year}-${month}-${day}`
```

Joonis 31. Faili loomise kuupäeva genereerimine.

E-arvete rakenduses on võimalik luua kolme eri tüüpi arveid: *Invoice*, *Invoice-Waybill* ja *Credit Invoice*. Valitud tüüp määrab ära, kas tegemist on deebetarve (DEB) või

kreeditarvega (CRE) ehk milline tähtede kombinatsioon kirjutatakse elemendi *Type* atribuuti *type*. Joonis 32 kirjeldab arve tüübi määramist, st *Invoice* ja *Invoice-Waybill* puhul on tegemist deebetarve, *Credit Invoice* puhul kreeditarvega.

```
let type
if (
this.invoice.invoiceType === 'Invoice' ||
this.invoice.invoiceType === 'Invoice-Waybill'
) {
    type = 'DEB'
} else {
    type = 'CRE'
}
```

Joonis 32. Arve tüübi määramine.

Kui arve koostaja ehk müüja valuuta erineb arve saaja ehk ostja valuutast, mis on arvele märgitud, tuleb maksekorralduses maksmisele kuuluv summa ehk elemendi *PaymentTotalSum* sisu ümber konverteerida – seda funktsionaalsust kajastab Joonis 33. Valuutade konverteerimiseks on kasutatud välist teenust Foreign exchange rates API with currency conversion [25], mis võtab valuutakursid Euroopa Keskpanga koduleheküljelt.

```
let paymentTotalSum
if (this.invoice.currency !== this.company.currency) {
    const getExchangeRate = async () => {
        const response = await
        axios.get(`https://api.exchangeratesapi.io/latest?base=${this.company.currency}`)
        return
        response.data.rates[`${this.invoice.currency}`]
    }

    const exchangeRate = await getExchangeRate()
    paymentTotalSum = +(Math.round(this.invoice.total *
    exchangeRate + 'e+2') + 'e-2')
} else {
    paymentTotalSum = this.invoice.total
}
```

Joonis 33. Tasumisele kuuluva summa konverteerimine.

Joonis 34 kirjeldab e-arve JSON objekti moodustamist, kusjuures tuleb rõhutada, et oluline on, et elemendid oleks JSON-is ka õiges järjekorras, et need XML-iks ümber konverteerides samamoodi õiges järjekorras paikneksid ning XML valideeruks. XML-i valideerimist on lähemalt käsitletud peatükis 4.5.

```

let json = {
  _declaration: {
    _attributes: {
      version: '1.0',
      encoding: 'UTF-8'
    }
  },
  E_Invoice: {
    _attributes: {
      'xmlns:xsi': 'http://www.w3.org/2001/XMLSchema-
instance',
      'xsi:noNamespaceSchemaLocation': 'e-
invoice_ver1.2.xsd'
    },
    Header: {
      Date: {
        _text: invoiceDate
      },
      FileId: {
        _text: invoiceId
      },
      Version: {
        _text: '1.2'
      }
    },
    Invoice: {
      InvoiceParties: {
        SellerParty: {
          Name: {
            _text: this.company.name
          },
          RegNumber: {
            _text: this.company.registrationNo
          }
        },
        BuyerParty: {
          Name: {
            _text: this.invoice.customerName
          },
          RegNumber: {
            _text: this.invoice.registrationNo
          }
        }
      },
      InvoiceInformation: {
        Type: {
          _attributes: {
            type
          }
        },
        DocumentName: {

```

```

    _text: this.invoice.invoiceType
  },
  InvoiceNumber: {
    _text: this.invoice.invoiceNo
  },
  InvoiceDate: {
    _text: this.invoice.date
  },
  DueDate: {
    _text: this.invoice.dueDate
  },
},
InvoiceSumGroup: {
  InvoiceSum: {
    _text: this.invoice.totalWithoutVat
  },
  TotalSum: {
    _text: this.invoice.total
  }
},
InvoiceItem: {
  InvoiceItemGroup: []
},
PaymentInfo: [],
_attributes: {
  invoiceId: this.invoice.invoiceNo,
  regNumber: this.invoice.registrationNo,
  sellerRegnumber: this.company.registrationNo
}
},
Footer: {
  TotalNumberInvoices: {
    _text: 1
  },
  TotalAmount: {
    _text: this.invoice.total
  }
}
}
}
}

```

Joonis 34. E-arve JSON kujul.

Kuna tooteid võib arvel olla üks või rohkem, lisatakse tooted arvele *for...in loop*'iga (Joonis 35).



```

for (const p in products) {
  const product = products[p]
  json.E_Invoice.Invoice.InvoiceItem.InvoiceItemGroup[p] = {
    ItemEntry: {
      Description: {
        _text: product.productName
      },
    },
    ItemDetailInfo: {
      ItemUnit: {
        _text: product.unit || ''
      },
      ItemAmount: {
        _text: product.quantity || 1
      },
      ItemPrice: {
        _text: product.price
      },
    },
  },
  ItemSum: {
    _text: product.price * product.quantity
  },
  Addition: {
    AddContent: {
      _text: 'Discount'
    },
    AddRate: {
      _text: -product.discount
    },
    _attributes: {
      addCode: 'DSC'
    }
  },
  VAT: {
    VATRate: {
      _text: product.vat
    },
    VATSum: {
      _text: +(Math.round(product.totalWithoutVat
        * (product.vat / 100))+ 'e+2') + 'e-2')
    },
    Currency: {
      _text: this.invoice.currency
    }
  },
  ItemTotal: {
    _text: product.total
  },
}
}

```

Joonis 35. Toodete lisamine elementi *InvoiceItemGroup*.

E-arve standardi kohaselt on maksekorralduse ehk *PaymentInfo* ploki kohustuslik kajastada kas elementi *PaymentRefId* ehk viitenumbrit või elementi *PaymentDescription* ehk maksekorralduse selgitust. See, kumb XML-i lisatakse, sõltub E-arvete rakenduses sellest, kas viitenumbri ehk *Reference number* väli on arvevormis täidetud – kui on lisatakse XML-i element *PaymentRefId* selle väärtusega, mis arvevormil kirjas, kui mitte, lisatakse element *PaymentDescription*, mille sisuks on arve tüüp koos arvenumbriga, mida näitab ka Joonis 36.

```
let paymentDescrOrRefId
if (this.invoice.referenceNo) {
  paymentDescrOrRefId = {
    PaymentRefId: {
      _text: this.invoice.referenceNo
    },
  }
} else {
  paymentDescrOrRefId = {
    PaymentDescription: {
      _text: `${this.invoice.invoiceType} number
      ${this.invoice.invoiceNo}`
    },
  }
}
```

Joonis 36. Elemendi *PaymentRefId* või *PaymentDescription* lisamine e-arve JSON objekti

Joonisel 37 kirjeldatakse maksekorralduse ploki ehk *PaymentInfo* elemendi moodustamist ning selle lisamist e-arve JSON objekti.

```

const paymentInfoElement = {
  Currency: {
    _text: this.company.currency
  },
  ...paymentDescrOrRefId,
  Payable: {
    _text: 'YES'
  },
  PayDueDate: {
    _text: this.invoice.dueDate
  },
  PaymentTotalSum: {
    _text: paymentTotalSum
  },
  PayerName: {
    _text: this.invoice.customerName
  },
  PaymentId: {
    _text: this.invoice.invoiceNo
  },
  PayToAccount: {
    _text: this.company.iban
  },
  PayToName: {
    _text: this.company.name
  },
}
json.E_Invoice.Invoice.PaymentInfo.push(paymentInfoElement)

```

Joonis 37. Maksekorralduse andmete lisamine e-arve JSON objekti.

JSON-ist XML-i konverteerimiseks kasutas autor npm-i *library*'t xml-js [22], mille abil saab ka erimärgid (&, ', >, <, ") ära kodeerida (Joonis 38).

```

const options = { compact: true, ignoreComment: true,
spaces: 4, textFn: (value) => value.replace(/'/g,
'&apos;').replace(/"/g, '&quot;')}
let xml = convert.js2xml(json, options)
this.invoiceXml = xml

```

Joonis 38. Erimärgide kodeerimine ja JSON-i XML-iks konverteerimine.

Pärast konverteerimist väärtustatakse loodud XML muutujale invoiceXml, mida kasutatakse XML faili loomiseks (täpsemalt kirjeldatud peatükis 4.4.5). Genereeritud XML-i struktuuri saab näha Lisas 13.

#### 4.4.5 XML failide salvestamine API-sse

Käesolev peatükk kirjeldab konkreetselt E-arvete rakenduses loodud arvetega seotud failide salvestamist, *app*'i üleslaetavate arvete puhul on süsteem veidi erinev. Loodud arvete XML failide salvestamiseks on New Invoice vaate aluseks olevas komponendis *new.vue* loodud meetod `uploadInvoiceFile`, mida kirjeldab Joonis 39. Alustuseks genereeritakse faili XML kood, mida on käsitletud pikemalt peatükis 4.4.4, seejärel luuakse objekt *file*, millele omistatakse *property*'d *fileName* ehk failinimi, *file* ehk faili sisu, milleks on loodud arve XML kood, ja *invoice* ehk arve ID, et seostada salvestatav fail kindla arvega. Viimaseks käivitatakse `uploadFile` funktsioon, millele pannakse parameetriga *file* kaasa faili andmed ja parameeter *url* viitab, millisele tabelile tuleb päring teha.

```
uploadInvoiceFile() {
  let savedInvoice = this.getSavedInvoice()
  if (savedInvoice) {
    this.generateXML(savedInvoice.id, this.products)
    const file = {
      fileName: `${this.invoice.invoiceType}
        ${this.invoice.invoiceNo}`,
      file: this.invoiceXml,
      invoice: savedInvoice.id
    }
    this.uploadFile({
      file: file, callback: null, url: 'createdFiles'
    })
  }
}
```

Joonis 39. `uploadInvoiceFile` meetod.

`uploadFile` on üks *ErplyApp.js* Vuex *store action*'itest ning seda kirjeldav kood asub Joonisel 40.

```
uploadFile({ }, { file, callback, url }) {
  const FileApi = new
    ErplyCloudApi(process.env.apiUrl[url])
  FileApi.CREATE(file, () => {
    this.loading = false
    if (callback) callback()
  })
}
```

Joonis 40. `uploadFile` Vuex *store action*.

Meetodi uploadInvoiceFile tulemusena on CREATED\_FILES API tabelisse loodud uus kirje, mis on seotud ühe kindla CREATED\_INVOICES tabelis oleva e-arvega arve ID järgi.

#### **4.4.6 XML faili allalaadimine API-st**

E-arvete rakenduse üks olulistest funktsionaalsustest on ka arve XML faili allalaadimine. Selles peatükis on välja toodud, kuidas see toimib rakendusse üleslaetud arvete puhul ehk Received Invoices vaates. XML failide allalaadimiseks on kaks varianti – on võimalik laadida alla ühe arve andmeid sisaldav XML fail märkides linnukesega ära ühe arve või moodustada *bulk* XML fail märkides ära 2 või rohkem arvet. Vajutades nupule „Download XML“ läheb käima meetod downloadXml (Joonis 41).

```

downloadXml() {
  if (this.selectedRows.length > 1) {
    let invoicesArray = []
    let invoicesSum = 0

    this.selectedRows.forEach(row => {
      const xmlFile = this.files.filter(file => file.invoice
      === row.id)

      const data = xmlFile[0].file
      const startIndex = data.search('<Invoice')
      const endIndex = data.search('<Footer>')
      const invoiceData = data.substring(startIndex, endIndex)
      invoicesArray.push(invoiceData)

      const totalSumStartIndex =
      data.search('<TotalSum>') + '<TotalSum>'.length
      const totalSumEndIndex = data.search('</TotalSum>')
      const invoiceSum = parseFloat(
      data.substring(totalSumStartIndex, totalSumEndIndex)
      )
      invoicesSum += invoiceSum
    })

    const invoicesTotal = parseFloat(invoicesSum).toFixed(2)
    const invoices = invoicesArray.join('')

    const bulkXML = this.generateBulkXML(
      invoices,
      invoicesArray.length,
      invoicesTotal
    )
    this.download('payment', bulkXML)
  } else if (this.selectedRows.length === 1) {
    const xmlFile = this.files.filter(
      file => file.invoice === this.selectedRows[0].id
    )
    const filename = xmlFile[0].fileName
    const fileData = xmlFile[0].file
    this.download(filename, fileData)
  }
}

```

Joonis 41. downloadXml meetod.

Kui märgistatud ridu on 1, laetakse andmebaasist alla täpselt see sama arvega seotud fail, mis on eelnevalt saabunud arvete failide tabelisse üles laetud ehk download meetodile antakse kaasa ühe konkreetse faili andmed ja nimi. Kui aga märgitud on rohkem ridu, selekteeritakse valitud arvete XML koodist välja elemendi *Invoice* osa ning lisatakse need

arvete listi, leitakse iga arve kogusumma ja summa ilma maksudeta ning liidetakse vastavalt omavahel kokku kõik arvete kogusummad ja kõik arvete summad ilma maksudeta. Seejärel käivitatakse funktsioon `generateBulkXml` (Joonis 42), millele antakse parameetritena kaasa arvete andmed, arvete hulk ja arvete kogusumma.

```
generateBulkXML(invoices, noOfInvoices, invoicesTotal) {  
    ...  
    const xml = `  
        <?xml version="1.0" encoding="UTF-8"?>  
        <E_Invoice xsi:noNamespaceSchemaLocation="e-  
            invoice_ver1.2.xsd"  
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-  
            instance">  
            <Header>  
            <Date>${invoiceDate}</Date>  
            <FileId>1</FileId>  
            <Version>1.2</Version>  
            </Header>  
            ${invoices}  
            <Footer>  
            <TotalNumberInvoices>${noOfInvoices}</TotalNumberIn  
            voices>  
            <TotalAmount>${invoicesTotal}</TotalAmount>  
            </Footer>  
            </E_Invoice>  
        `;  
    return xml  
}
```

Joonis 42. `generateBulkXml` meetod.

Tulemuseks on *bulk* XML faili kood, mis edastatakse parameetrina meetodile `download` (Joonis 43). Kuna arve genereerimise kuupäeva ehk *Date* elemendi sisu loomine kordub Joonisel 31 väljatooduga, on see osa Jooniselt 42 välja jäetud ning asendatud kolme punktiga.

```

download(filename, file) {
  let element = document.createElement('a')
  element.setAttribute(
    'href',
    'data:text/xml;charset=utf-8,' +
    encodeURIComponent(file)
  )
  element.setAttribute('download', filename)
  element.style.display = 'none'
  document.body.appendChild(element)
  element.click()
  document.body.removeChild(element)
}

```

Joonis 43. download meetod.

Meetodi download tulemusena laetakse XML fail rakenduse kasutaja seadmesse. Ühte arvet sisaldava faili korral on faili nimeks arve tüüp koos arve numbriga, *bulk* XML korral on allalaetava faili nimeks „payment“.

#### 4.4.7 Arvete kustutamine

Arvete kustutamine on üks Created Invoices tabeli funktsionaalsustest. Selleks, et arveid kustutada, tuleb linnukesega märkida sobivad read ning vajutada prügikasti kujutisele, mis käivitab deleteInvoice meetodi, mida on kajastatud Joonisel 44.

```

deleteInvoice() {
  this.selectedRows.forEach(invoice => {
    this.deleteRecord({
      record: invoice,
      callback: null,
      url: 'createdInvoices'
    })
  })
}

```

Joonis 44. deleteInvoice meetod.

Meetodis kutsutakse välja Vuex *store*'i *action* deleteRecord (Joonis 45), millele antakse kaasa kustutatava arve andmed ning *url*, et määrata, millisele tabelile tuleks päring teha.



```

deleteRecord({ commit }, { record, callback, url }) {
  record.deletedAt = new Date().toISOString()
  const InvoicesApi = new
  ErplyCloudApi(process.env.apiUrl[url])
  InvoicesApi.DELETE(record.id, () => {
    commit("deleteLocalRecord", record.id)
    if (callback) callback()
  })
}

```

Joonis 45. deleteRecord Vuex store action.

Meetod DELETE ei ole midagi muud, kui arvete tabelile tehtav *axios*'e delete päring (Joonis 46).

```

DELETE(rowId, callback) {
  HTTP.delete(this.endpointBase + rowId).then(() => {
    if (callback) callback()
  })
}

```

Joonis 46. DELETE meetod.

deleteLocalRecord *mutation* (Joonis 47) on vajalik selleks, et ka lokaalselt, *state*'ist arve ära kustutada ja UI-d uuendada.

```

deleteLocalRecord(state, recordId) {
  const index = state.records.findIndex(r => r.id ===
  recordId)
  state.records.splice(index, 1)
},

```

Joonis 47. deleteLocalRecord Vuex store mutatsioon.

deleteInvoice meetodi tulemusena on märgitud kirje või kirjed nii CREATED\_INVOICES API tabelist kui ka lokaalsest arvete tabelist kustutatud.

## 4.5 Loodud e-arve standardile vastavuse kontrollimine

Rahandusministri 11.04.2017 määruse nr 24 „Masintöödeldava algdokumendi juhendi kehtestamine“ lisa „E-arvete juhend“ [1] dokumendi lõpust leiab XML skeemi ehk XSD, millele Eesti e-arve standardi kohaselt e-arded vastama peavad.

XML skeem määrab ära XML dokumendis esinevad elemendid ja atribuudid, nende andmetüübid ja vaikumisi või fikseeritud väärtused ning alamelementide kogused. Samuti paneb XML skeem paika, millises järjekorras elemendid paiknema peavad [26].

Selleks, et veenduda E-arvete rakenduses loodavate XML failide korrektsuses, kasutas autor XML-i valideerimiseks veebis leiduvat XML validaatorit [27]. Valideerimise tulemus oli positiivne – E-arvete rakenduses loodavad e-arved vastavad nõuetele ning eksimusi ei esinenud.

## 5 Kokkuvõte

Käesoleva töö eesmärgiks oli luua esialgne versioon E-arvete rakendusest ettevõtte Erply sisendi põhjal, mis oleks suuteline Eesti e-arve standardile vastavaid e-arveid looma ning olemasolevaid e-arveid PDF-kujul vormindama.

Töö tulemusena valmis *app*, mis võimaldab kasutajal koostada e-arveid, mujalt saabunud e-arveid rakendusse üles laadida ning neid PDF-kujul vaadata. Rakenduse esimene versioon on inglisekeelne ning kasutatav laua- ja sülearvutites.

Et rakenduse abil loodavad failid vastaksid nõuetele, analüüsis töö autor süvitsi Eesti e-arve standardit ning järgis sealseid eeskirju rakenduse arendamisel. Lisaks tutvus autor e-arvete teoreetilise tausta, e-arvete vastuvõtjate registri ja e-arvete operaatoritega ning kirjeldas neid ja Eesti e-arve standardit käesolevas töös. Samuti on töös räägitud rakenduse arendamisel kasutatud olulisematest tehnoloogiatest.

Autor andis töös ülevaate ka kõigist rakenduse vaadetest ning tegevustest, mida erinevates vaadetes teha on võimalik. Olulisemateks loodud funktsionaalsusteks võib pidada Actual Reports-i abil e-arvete PDF-kujul kuvamist; uute arvete loomist ja arvele lisatud toodete salvestamist; e-arve XML failide loomist ja salvestamist andmebaasi ning arvete kustutamist.

Testimiseks kasutas autor veebis leiduvat XML validaatorit – rakenduse abil genereeritud XML fail valideerus. Seega võib väita, et E-arvete rakenduses loodavad e-arved vastavad Eesti e-arve standardile. Samuti on rakendus võimeline kuvama üleslaetavaid faile PDF-kujul, seega said kõik algselt püstitatud eesmärgid täidetud.

Töö autori arvates on Vue.js ja Nuxt.js väga sobilikud ja mugavad taoliste veebirakenduste loomiseks, seetõttu on kindlasti plaanis rakendust ka edaspidi just neid raamistikke kasutades täiustada, kasutajasõbralikumaks muuta ja rakendusse uusi funktsionaalsusi lisada.

## Kasutatud kirjandus

- [1] Riigiteataja, „E-arve juhend,“ 11. aprill 2017. [Võrgumaterjal]. Available: [https://www.riigiteataja.ee/aktilisa/1130/4201/7005/RM\\_m24\\_lisa.pdf](https://www.riigiteataja.ee/aktilisa/1130/4201/7005/RM_m24_lisa.pdf). [Kasutatud 4. veebruar 2019].
- [2] Technopedia Inc., „Base64,“ [Võrgumaterjal]. Available: <https://www.techopedia.com/definition/27209/base64>. [Kasutatud 20. mai 2019].
- [3] w3schools.com, „What is Command Line Interface (CLI)?,“ [Võrgumaterjal]. Available: [https://www.w3schools.com/whatis/whatis\\_cli.asp](https://www.w3schools.com/whatis/whatis_cli.asp). [Kasutatud 5. mai 2019].
- [4] Rahandusministeerium, „E-arded,“ 25. veebruar 2019. [Võrgumaterjal]. Available: <https://www.rahandusministeerium.ee/et/riigihaldus/e-arded>. [Kasutatud 27. aprill 2019].
- [5] I. Kald, „Riik läheb 1. juulist üle vaid e-arvetele,“ ITuudised.ee, 11. jaanuar 2019. [Võrgumaterjal]. Available: <https://www.ituudised.ee/uudised/2019/01/11/riik-laheb-1-juulist-ule-vaid-e-arvetele->. [Kasutatud 23. aprill 2019].
- [6] M. Liiv, „E-arded on tõusev trend,“ Raamatupidaja.ee, 18. märts 2019. [Võrgumaterjal]. Available: <https://www.raamatupidaja.ee/uudised/2019/03/18/standard-booksi-kasutajate-seas-on-e-arded-tousev-trend>. [Kasutatud 27. aprill 2019].
- [7] S. Tiits, „Mille poolest e-arvete operaatorid erinevad?,“ Omniva Arvekeskus, 26. märts 2019. [Võrgumaterjal]. Available: <https://www.arvekeskus.ee/mille-poolest-e-arvete-operaatorid-erinevad/>. [Kasutatud 28. aprill 2019].
- [8] Omniva, „Arvekeskusest,“ [Võrgumaterjal]. Available: <https://www.omniva.ee/ari/arvekeskus/tutvustus>. [Kasutatud 27. aprill 2019].
- [9] Fitek AS, „E-arvete keskkond,“ [Võrgumaterjal]. Available: <https://fitek.ee/e-arvete-keskkond/>. [Kasutatud 28. aprill 2019].
- [10] Fitek AS, „Fitek Premium,“ [Võrgumaterjal]. Available: [https://www.arved.ee/public/co\\_fitek.html](https://www.arved.ee/public/co_fitek.html). [Kasutatud 28. aprill 2019].
- [11] Omniva, „Lahendused,“ [Võrgumaterjal]. Available: <https://www.omniva.ee/ari/arvekeskus/lahendused>. [Kasutatud 19. mai 2019].
- [12] Amphora Professional, „E-arve,“ [Võrgumaterjal]. Available: <https://www.amphora.ee/default.aspx?menu=3025&loc=03>. [Kasutatud 19. mai 2019].
- [13] Registrate ja Infosüsteemide Keskus, „e-arveldaja,“ [Võrgumaterjal]. Available: <https://www.rik.ee/et/e-arveldaja>. [Kasutatud 28. aprill 2019].
- [14] Telema, „eInvoicing - edumeelsed saadavad e-arveid,“ [Võrgumaterjal]. Available: <https://telema.ee/einvoicing/>. [Kasutatud 19. mai 2019].
- [15] Vue.js, „Introduction,“ [Võrgumaterjal]. Available: <https://vuejs.org/v2/guide/>. [Kasutatud 3. mai 2019].

- [16] Nuxt.js, „Introduction,“ [Võrgumaterjal]. Available: <https://nuxtjs.org/guide/>. [Kasutatud 3. mai 2019].
- [17] webpack, „Hot Module Replacement,“ [Võrgumaterjal]. Available: <https://webpack.js.org/concepts/hot-module-replacement/>. [Kasutatud 3. mai 2019].
- [18] Vuetify, „Quick start,“ [Võrgumaterjal]. Available: <https://vuetifyjs.com/en/getting-started/quick-start>. [Kasutatud 4. mai 2019].
- [19] Vue Material, „Vue Material - Getting Started,“ [Võrgumaterjal]. Available: <https://vuematerial.io/getting-started/>. [Kasutatud 4. mai 2019].
- [20] npm, Inc., „About npm,“ [Võrgumaterjal]. Available: <https://www.npmjs.com/about>. [Kasutatud 5. mai 2019].
- [21] npm, Inc., „About npm,“ [Võrgumaterjal]. Available: <https://docs.npmjs.com/about-npm/>. [Kasutatud 5. mai 2019].
- [22] „xml-js,“ [Võrgumaterjal]. Available: <https://www.npmjs.com/package/xml-js>. [Kasutatud 25. veebruar 2019].
- [23] Actual Reports, „A Cloud Based Document Generation Platform,“ [Võrgumaterjal]. Available: <https://actualreports.com/>. [Kasutatud 2. mai 2019].
- [24] Pluralsight, [Võrgumaterjal]. Available: <https://www.pluralsight.com/>. [Kasutatud detsember 2019].
- [25] „Foreign exchange rates API with currency conversion,“ [Võrgumaterjal]. Available: <https://exchangeratesapi.io/>. [Kasutatud 14. mai 2019].
- [26] w3schools.com, „XML Schema Tutorial,“ [Võrgumaterjal]. Available: [https://www.w3schools.com/xml/schema\\_intro.asp](https://www.w3schools.com/xml/schema_intro.asp). [Kasutatud 17. mai 2019].
- [27] freeformatter.com, „XML Validator - XSD (XML Schema),“ [Võrgumaterjal]. Available: <https://www.freeformatter.com/xml-validator-xsd.html>. [Kasutatud 14. mai 2019].

# Lisa 1 – E-arvete rakenduse kasutusjuhud



## Lisa 2 – Received Invoices vaade

E-Invoicing

Received Invoices Created Invoices New Invoice Company Details

DOWNLOAD XML

Search

<input type="checkbox"/>	Date	Invoice type	Invoice no.	From	Total
<input type="checkbox"/>	2019-05-15	Invoice	221	Näidis Ettevõtte AS	70.81 EUR
<input type="checkbox"/>	2013-12-01	ARVE	1234	TESTMÜÜJA AS	1.20 EUR
<input type="checkbox"/>	2019-05-06	Invoice-Waybill	77114	Näidis Ettevõtte AS	616.74 EUR
<input type="checkbox"/>	2019-05-15	Invoice	7871	Näidis Ettevõtte AS	14.19 EUR
<input type="checkbox"/>	2019-05-15	Invoice	5465	Näidis Ettevõtte AS	11.99 EUR

Rows per page: 10 1-5 of 5

+

## Lisa 3 – Received Invoices View vaade

E-Invoicing

Received Invoices Created Invoices New Invoice Company Details

DOWNLOAD XML Search

<input type="checkbox"/>	Date	Invoice type	Invoice no.	From	Total
<input type="checkbox"/>	2019-05-15	Invoice	221	Näidis Ettevõtte AS	70.81 EUR
<input type="checkbox"/>	2013-12-01	ARVE	1234	TESTMÜÜJA AS	1.20 EUR
<input type="checkbox"/>	2019-05-06	Invoice-Waybill	77114	Näidis Ettevõtte AS	616.74 EUR
<input type="checkbox"/>	2019-05-15	Invoice	7871	Näidis Ettevõtte AS	14.19 EUR
<input type="checkbox"/>	2019-05-15	Invoice	5465	Näidis Ettevõtte AS	11.99 EUR

Rows per page: 10 1-5 of 5

+

### Invoice-Waybill 77114

Company: Company AS Date: 2019-05-06  
 Registration number: 798798 Due date: 2019-05-17  
 IBAN: EE12345678912345678921787 Currency: EUR

Product/Service	Price	Qty	Unit	Disc	VAT	Total
8798	541.00	1	.	5%	20%	616.74

Total (without VAT): 513.95  
 Total (EUR): 616.74



## Lisa 4 – Upload vaade enne faili üleslaadimist

The screenshot displays an e-invoicing application interface. At the top, there is a navigation bar with the following tabs: "Received Invoices" (active), "Created Invoices", "New Invoice", and "Company Details". A search bar is located at the top right of the main content area, with the label "Search" and a magnifying glass icon. Below the search bar is a table of received invoices. The table has the following columns: "Date", "Invoice type", "Invoice no.", "From", and "Total". The table contains five rows of data. To the right of the table is a large light blue dashed box with the text "Drag your file here to begin or click to browse". Below the table, there is a pagination control showing "Rows per page: 10" and "1-5 of 5". A blue circular button with a white plus sign is located at the bottom right of the interface.

<input type="checkbox"/>	Date	Invoice type	Invoice no.	From	Total
<input type="checkbox"/>	2019-05-15	Invoice	221	Näidis Ettevõtte AS	70.81 EUR
<input type="checkbox"/>	2013-12-01	ARVE	1234	TESTMÜÜJA AS	1.20 EUR
<input type="checkbox"/>	2019-05-06	Invoice-Waybill	77114	Näidis Ettevõtte AS	616.74 EUR
<input type="checkbox"/>	2019-05-15	Invoice	7871	Näidis Ettevõtte AS	14.19 EUR
<input type="checkbox"/>	2019-05-15	Invoice	5465	Näidis Ettevõtte AS	11.99 EUR

## Lisa 5 – Upload vaade pärast faili üleslaadimist

The screenshot displays the 'E-Invoicing' software interface. The main navigation bar includes 'Received Invoices', 'Created Invoices', 'New Invoice', and 'Company Details'. A search bar is located at the top of the main content area. Below the search bar is a table listing received invoices. The table has columns for 'Date', 'Invoice type', 'Invoice no.', 'From', and 'Total'. There are five rows of data, each with a checkbox in the first column. Below the table, there is a pagination control showing 'Rows per page: 10' and '1-5 of 5'. To the right of the table is a modal window titled 'Invoice-Waybill 878'. This modal contains the following information:

**Invoice-Waybill 878**

Company: Company AS      Date: 2019-05-15  
Registration number: 79798      Due date: 2019-05-29  
IBAN: EE12345678912345678921787      Currency: EUR

Product/Service	Price	Qty	Unit	Disc	VAT	Total
Product	47.89	2	-	5%	20%	109.19

Total (without VAT): 90.99  
Total (EUR): 109.19

## Lisa 6 – Created Invoices vaade

E-Invoicing

Received Invoices **Created Invoices** New Invoice Company Details

SEND [trash] [download] Search [magnifying glass]




<input type="checkbox"/>	Created at	Invoice type	Company name	Invoice no.	Total	Status
<input type="checkbox"/>	2019-05-07	Invoice	Erply123456789	321010	255.51 EUR	● Saved
<input type="checkbox"/>	2019-05-07	Invoice	Test Company AS	447148	88.92 SEK	● Saved
<input type="checkbox"/>	2019-05-07	Invoice	Company AS	1155-88	34.20 BGN	● Saved
<input type="checkbox"/>	2019-05-07	Invoice	Erply123456789	222-11-45	2.00 EUR	● Saved
<input type="checkbox"/>	2019-05-07	Invoice-Waybill	Erply123456789	65654	4.56 EUR	● Saved
<input type="checkbox"/>	2019-05-07	Credit Invoice	Company 111	71484	36.72 EUR	● Saved
<input type="checkbox"/>	2019-05-06	Invoice	Test	7884454	10.89 EUR	● Saved
<input type="checkbox"/>	2019-05-06	Invoice	Erply123456789	124124444	1.00 EUR	● Saved
<input type="checkbox"/>	2019-05-06	Invoice	Company AS	1234567811	20766.24 EUR	● Saved

[plus]

# Lisa 7 – Created Invoices Edit vaade

E-Invoicing

Received Invoices **Created Invoices** New Invoice Company Details

SEND   Search 

<input type="checkbox"/>	Created at	Invoice type	Company name	Invoice no.
<input type="checkbox"/>	2019-05-07	Invoice	Erply123456789	321010
<input type="checkbox"/>	2019-05-07	Invoice	Test Company AS	447148
<input type="checkbox"/>	2019-05-07	Invoice	Company AS	1155-88
<input type="checkbox"/>	2019-05-07	Invoice	Erply123456789	222-11-45
<input type="checkbox"/>	2019-05-07	Invoice-Waybill	Erply123456789	65654
<input type="checkbox"/>	2019-05-07	Credit invoice	Company 111	71484
<input type="checkbox"/>	2019-05-06	Invoice	Test	7884454
<input type="checkbox"/>	2019-05-06	Invoice	Erply123456789	12412444

**+**

Invoice type \*  
Invoice

Company name \*  
Erply123456789

Registration number \*  
46546456546184

IBAN \*  
EE123456789123456789

Reference number

Invoice no. \*  
321010

Date  
2019-05-07

Due date  
2019-05-21

Currency \*  
EUR

SAVE

Product/Service	Price	Qty	Unit	Disc.	VAT	Total
Test #2	98.74	2	-	4 %	9 %	206.64

# Lisa 8 – Created Invoices View vaade

**Ei** E-Invoicing

Received Invoices **Created Invoices** New Invoice Company Details

SEND    Search

<input type="checkbox"/>	Created at	Invoice type	Company name	Invoice no.
<input type="checkbox"/>	2019-05-07	Invoice	Erply123456789	321010
<input type="checkbox"/>	2019-05-07	Invoice	Test Company AS	447148
<input type="checkbox"/>	2019-05-07	Invoice	Company AS	1155-88
<input type="checkbox"/>	2019-05-07	Invoice	Erply123456789	222-11-45
<input type="checkbox"/>	2019-05-07	Invoice-Waybill	Erply123456789	65654
<input type="checkbox"/>	2019-05-07	Credit invoice	Company 111	71484
<input type="checkbox"/>	2019-05-06	Invoice	Test	788445
<input type="checkbox"/>	2019-05-06	Invoice	Erply123456789	124124444

**Invoice 124124**

Company: Erply123456789 Date: 2019-04-10  
Registration number: 46546456546184 Due date: 2019-04-24  
IBAN: EE123456789123456789 Currency: EUR

Product/Service	Price	Qty	Unit	Disc	VAT	Total
Test 1	12.00	1	.	5%	20%	13.68

Total (without VAT): 11.40  
Total (EUR): 13.68

# Lisa 9 – New Invoice vaade

**E-Invoicing**

Received Invoices Created Invoices **New Invoice** Company Details

SEND [trash] [download] Search [magnifying glass]

<input type="checkbox"/>	Created at	Invoice type	Company name	Invoice no.
<input type="checkbox"/>	2019-05-07	Invoice	Erply123456789	321010
<input type="checkbox"/>	2019-05-07	Invoice	Test Company AS	447148
<input type="checkbox"/>	2019-05-07	Invoice	Company AS	1155-88
<input type="checkbox"/>	2019-05-07	Invoice	Erply123456789	222-11-45
<input type="checkbox"/>	2019-05-07	Invoice-Waybill	Erply123456789	65654
<input type="checkbox"/>	2019-05-07	Credit invoice	Company 111	71484
<input type="checkbox"/>	2019-05-06	Invoice	Test	7884454
<input type="checkbox"/>	2019-05-06	Invoice	Erply123456789	124124444

Invoice type\* [dropdown] Invoice no.\* [text]

Company name\* [text] Date [calendar]

Registration number\* [text] Due date [calendar]

IBAN\* [text] Currency\* [dropdown]

Reference number [text]


+ [plus icon]

Product/Service	Price	Qty	Unit	Disc.	VAT	Total
						Total (without VAT): 0.00

SAVE

## Lisa 10 – Company Details vaade

E-Invoicing

Received Invoices Created Invoices New Invoice **Company Details** 

Company name\*  
Majandustarkvara OÜ SAVE

IBAN\*  
EE123456789123456789

Registration number\*  
654987351324481

Currency\*  
EUR ▼

# Lisa 11 – E-arvete API tabelid

Create application

? x

Overview

Dependencies

Model

Errors

Advanced controls

Custom tables

COMPANY

CREATED\_FILES

CREATED\_INVOICES

PRODUCTS

RECEIVED\_FILES

RECEIVED\_INVOICES

Source app

e-invoicing-api (v0.1.0)

Source app UUID

00000001-5fba723-jrg44p98

DB table name

CREATED\_FILES

Fields

API

?

1. ID PK (Auto increment)
2. CREATED\_AT Trigger (Created at)
3. UPDATED\_AT Trigger (Updated at)
4. CREATED\_BY Trigger (Created by)
5. UPDATED\_BY Trigger (Updated by)
6. FILE\_NAME VARCHAR(255)
7. FILE LONGTEXT
8. INVOICE Ref-cascade (CREATE...OICES, ID)

Close



# Lisa 12 – Võimalikud päringud E-arvete rakenduse API tabelile

**e-invoicing-app API** 1.0.0 GA 50  
api/00000001-17ad1a16-jrao7mad/openapi.json

Servers  
 Authorize

Filter by tag

**created-invoices** ▼

- GET** /created-invoices Query records from CREATED\_INVOICES
- POST** /created-invoices Add new record to CREATED\_INVOICES
- GET** /created-invoices/{recordId} Query one record from CREATED\_INVOICES
- PUT** /created-invoices/{recordId} Modify one record in CREATED\_INVOICES
- DELETE** /created-invoices/{recordId} Delete one record in CREATED\_INVOICES
- PUT** /created-invoices/bulk Modify multiple records in CREATED\_INVOICES
- POST** /created-invoices/bulk Add multiple new records to CREATED\_INVOICES

## Lisa 13 – E-arvete rakenduse poolt genereeritud XML faili

### näidis

```
<?xml version="1.0" encoding="UTF-8"?>
<E_Invoice xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="e-invoice_ver1.2.xsd">
  <Header>
    <Date>2019-05-14</Date>
    <FileId>83</FileId>
    <Version>1.2</Version>
  </Header>
  <Invoice invoiceId="54548" regNumber="87871"
sellerRegnumber="65498735132">
    <InvoiceParties>
      <SellerParty>
        <Name>Näidis Ettevõtte AS</Name>
        <RegNumber>65498735132</RegNumber>
      </SellerParty>
      <BuyerParty>
        <Name>Test4545</Name>
        <RegNumber>87871</RegNumber>
      </BuyerParty>
    </InvoiceParties>
    <InvoiceInformation>
      <Type type="DEB"/>
      <DocumentName>Invoice</DocumentName>
      <InvoiceNumber>54548</InvoiceNumber>
      <InvoiceDate>2019-05-14</InvoiceDate>
      <DueDate>2019-05-28</DueDate>
    </InvoiceInformation>
    <InvoiceSumGroup>
      <InvoiceSum>23.77</InvoiceSum>
      <TotalSum>26.15</TotalSum>
    </InvoiceSumGroup>
    <InvoiceItem>
      <InvoiceItemGroup>
        <ItemEntry>
          <Description>Product12</Description>
          <ItemDetailInfo>
            <ItemUnit>kg</ItemUnit>
            <ItemAmount>2</ItemAmount>
            <ItemPrice>12.51</ItemPrice>
          </ItemDetailInfo>
          <ItemSum>25.02</ItemSum>
          <Addition addCode="DSC">
            <AddContent>Discount</AddContent>
            <AddRate>-5</AddRate>
          </Addition>
          <VAT>
            <VATRate>10</VATRate>
            <VATSum>2.38</VATSum>
            <Currency>EUR</Currency>
          </VAT>
          <ItemTotal>26.15</ItemTotal>
        </ItemEntry>
      </InvoiceItemGroup>
    </InvoiceItem>
    <PaymentInfo>
```

```
<Currency>EUR</Currency>
      <PaymentRefId>778811254</PaymentRefId>
<Payable>YES</Payable>
<PayDueDate>2019-05-28</PayDueDate>
<PaymentTotalSum>26.15</PaymentTotalSum>
<PayerName>Test4545</PayerName>
<PaymentId>54548</PaymentId>

<PayToAccount>EE12345678912345678921787</PayToAccount>
      <PayToName>Näidis Ettevõte AS</PayToName>
    </PaymentInfo>
  </Invoice>
  <Footer>
    <TotalNumberInvoices>1</TotalNumberInvoices>
    <TotalAmount>26.15</TotalAmount>
  </Footer>
</E_Invoice>
```