

**VALSTS INFORMĀCIJAS SISTĒMU SAVIETOTĀJS
(VISS) UN VIENOTĀ VALSTS UN PAŠVALDĪBU
PAKALPOJUMU PORTĀLA WWW.LATVIJA.LV
PILNVEIDOŠANA UN UZTURĒŠANA**

**MAKSĀJUMU MODULIS UN IEDZĪVOTĀJU MAKSĀJUMU
NODROŠINĀŠANA**

INTEGRĀCIJAS INSTRUKCIJA

VDAA-13_7_17_41 -VISS_2016-MM-II

08.10.2024 versija 2.27

DOKUMENTA IDENTIFIKĀCIJA

DOKUMENTA ID:	VDAA-13_7_17_41 -VISS_2016-MM-II-V2.27-08.10.2024
DOKUMENTA NOSAUKUMS:	Valsts informācijas sistēmu savietotājs (VISS) un Vienotā valsts un pašvaldību pakalpojumu portāla www.latvija.lv pilnveidošana un uzturēšana. Maksājumu modulis un ledzīvotāju maksājumu nodrošināšana Integrācijas instrukcija.
DOKUMENTA KODS:	VDAA-13_7_17_41 -VISS_2016-MM-II
VERSIJA:	Versija 2.27, Laidiens 08.10.2024 (saīsināti V2.27 08.10.2024)

SASKAŅOJUMI

ORGANIZĀCIJA	VĀRDS, UZVārds, AMATS	DATUMS	PARAKSTS
Valsts digitālās attīstības aģentūra	Pasūtītāja pārstāvis		
SIA "ABC Softwares"	M.Pētersons, projekta vadītājs no Izpildītāja puses	02.02.2021	
SIA "ZZ Dats"	E.Vasiļevskis, projekta vadītājs no Izpildītāja puses	02.02.2021	
SIA "ZZ Dats"	E.Vasiļevskis, Izstrādātāja pārstāvis	02.02.2021	

IZMAIŅU VĒSTURE

VERSIJA	DATUMS	APRAKSTS	ORGANIZĀCIJA	AUTORS
2.02	07.03.2013.	Izstrādāta Word formāta instrukcija	SIA "ZZ Dats"	M.Gūtmanis
2.03	11.04.2013.	Viekti precizējumi pēc programmatūras izstrādes	SIA "ZZ Dats"	M.Gūtmanis
2.04	29.08.2013.	Pievienoti piemēri PHP	SIA "ZZ Dats"	M.Gūtmanis
2.05	15.10.2013.	Papildinātas nodaļas 2.1, 2.2, 2.3 Dzēsta nodaļa 2.4	SIA "ZZ Dats"	M.Gūtmanis
2.06	11.12.2013.	Papildināta nodaļa 2.2, 2.2.2, 2.4, 2.5 Izveidotas nodaļas 2.6, 0	SIA "ZZ Dats"	M.Gūtmanis
2.07	07.03.2014.	Papildinātas nodaļas 2.1, 2.2, 2.3	SIA "ZZ Dats"	M.Gūtmanis
2.08	07.04.2015.	Pievienota nodaļa 0	SIA "ZZ Dats"	M.Gūtmanis
2.09	13.05.2015.	Papildināts maksājumu pieprasījuma statusu apraksts nodaļā 2.2.2 un 2.8	SIA "ZZ Dats"	M.Gūtmanis
2.10	11.10.2016.	Papildināts .NET piemērs nodaļā 2.2	SIA "ZZ Dats"	E.Vasiļevskis
2.11	18.01.2017.	Precizētas nodaļas 2.2, 2.2.2, 2.4, 2.5 un 2.8	SIA "ZZ Dats"	M.Gūtmanis
2.12	28.07.2020	Pievienota nodaļa 2.10	SIA "ZZ Dats"	M. Bambītis
2.13	22.09.2020	Papildināta 2.10 nodaļa	SIA "ZZ Dats"	M. Bambītis
2.14	25.09.2020	Papildināta 2.10 nodaļa	SIA "ZZ Dats"	M. Bambītis
2.15	28.10.2020	Papildināta 2.10 nodaļa	SIA "ZZ Dats"	M. Bambītis
2.16	03.11.2020	Labota 2.10.2. nodaļa	SIA "ZZ Dats"	M.Bambītis
2.17	02.02.2021	Labota 2.10. nodaļa	SIA "ZZ Dats"	M.Bambītis
2.18	04.06.2021	Labots piemērs 2.10.1 un 3.1 nodaļās	SIA "ZZ Dats"	M.Bambītis
2.19	06.08.2021	Papildinātas nodaļas 2.2, 2.3, 2.10.1	SIA "ZZ Dats"	M.Bambītis
2.20	15.12.2023	Papildinātas 2.2.2, 2.4, 2.5, 3.1 nodaļas	SIA "ZZ Dats"	M.Bambītis
2.21	31.01.2024	Papildināt DU "Informācijas Resursa 'E-pakalpojumu infrastruktūra v3' tehnoloģiju bāzes atjaunošana, uzlabojumi, lai nodrošinātu pieejamības (Accessibility) prasību izpildi, kā arī uzlabojumi, balstoties uz klientu ieteikumiem" ietvaros	SIA "ZZ Dats"	M.Bambītis
2.22	07.03.2024	Pievienota 2.8. nodaļa	SIA "ZZ Dats"	M.Bambītis
2.23	26.03.2024	Pievienots SOAP rēķinu servisa izsaukuma PHP piemērs, pievienota 2.1.1.8 nodaļa, pārstrukturēts dokuments	SIA "ZZ Dats"	M.Bambītis
2.24	13.05.2024	Labotas apmaksas procesa shēmas Labojumi saskaņā ar komentāriem	SIA "ZZ Dats"	M.Bambītis
2.25	01.07.2024		SIA "ZZ Dats"	M.Bambītis
2.26	06.09.2024	Precizēta 2.2. nodaļa	SIA "ZZ Dats"	M.Bambītis
2.27	08.10.2024	Pārstrukturēta 2.2 nodaļa, papildināti koda piemēri	SIA "ZZ Dats"	M.Bambītis

Satura rādītājs

1.1. Dokumenta nolūks	5
1.2. Darbības sfēra	5
1.3. Terminu un pieņemtie apzīmējumi	5
1.4. Saistība ar citiem dokumentiem	5
1.5. Dokumenta pārskats	6
2. IZMANTOŠANAS SCENĀRIJI	7
2.1. Portāla latvija.lv pakalpojumu apmaksas veikšana	7
2.1.1. MM izsaukumi, izmantojot Lvp.EservicePlatform.Backend.PaymentApi mikroservisu	8
2.1.1.1. Maksāšanas pieprasījuma izveidošana	8
2.1.1.2. Maksāšanas pieprasījuma statusa pārbaude	10
2.1.1.3. Maksāšanas pieprasījumu izgūšana	10
2.1.1.4. Maksājumu izgūšana	12
2.1.1.5. Maksājuma statusa pārbaude	13
2.1.1.6. Maksāšanas pieprasījumu statusu pārbaude un apmaksas veida izgūšana	14
2.1.1.7. Rēķina izveidošana	15
2.1.1.8. Maksāšanas pieprasījuma atcelšana	16
2.2. Ārēju pakalpojumu apmaksas veikšana	17
2.2.1. Maksāšanas pieprasījuma izveidošana	18
2.2.2. Maksāšanas pieprasījuma statusa pārbaude	21
2.2.3. Maksāšanas pieprasījumu izgūšana	22
2.2.4. Maksājumu izgūšana	23
2.2.5. Maksājuma statusa pārbaude	24
2.2.6. Rēķina izveidošana	25
2.2.7. Maksāšanas pieprasījuma atcelšana	26
2.3. Maksāšanas procesa tīmekļa saskarne	27
2.4. Maksāšanas pieprasījuma attaisnojuma dokumenta datnes izgūšana PDF formātā – REST serviss	28
2.5. Maksāšanas pieprasījuma attaisnojuma dokumenta datu izgūšana XML formātā – REST serviss	30
2.6. Maksāšanas pieprasījuma attaisnojuma dokumenta datnes izgūšana PDF formātā – SOAP serviss	32
2.7. Maksāšanas pieprasījuma attaisnojuma dokumenta datu izgūšana XML formātā – SOAP serviss	32
2.8. Maksāšanas pieprasījuma attaisnojuma dokumenta datu izgūšana PDF vai XML formātā – SOAP serviss ar sertifikāta autorizāciju bez lomu pārbaudes	32
3. PIELIKUMI	35
3.1. Maksāšanas pieprasījuma piemērs	35

Ievads

Valsts informācijas sistēmu savietotāja (turpmāk tekstā – VISS) Maksājumu modulis un ledzīvotāju maksājumu nodrošināšanas modulis (turpmāk tekstā – MM).

1.1. Dokumenta nolūks

Dokuments sniedz detalizētu tehnisku informāciju par Maksājumu moduļa un ledzīvotāju maksājumu nodrošināšanas integrācijas iespējām informācijas sistēmās ar mērķi nodrošināt apmaksas veikšanu dažādiem pakalpojumiem.

Šis dokuments ir projekta „Par Valsts informācijas sistēmu savietotāja, Latvijas valsts portāla www.latvija.lv un elektronisko pakalpojumu izstrāde un uzturēšana” nodaļas Maksājumu moduļa integrācijas instrukcijas un paraugi (moduļa numurs finanšu piedāvājumā - 1.2.3) nodevums.

Dotais dokuments paredzēts:

- Pasūtītājam realizācijas kontrolei;
- Ārējo IS izstrādātājiem integrācijas risinājuma izstrādei.

1.2. Darbības sfēra

Timekļa pakalpojumu izsaukumi tiek veikti izmantojot X.509 sertifikātu. Latvija.gov.lv pakalpojumos MM API izsaukumi tiek veikti izmantojot epakalpojumu ietvara PaymentApi klasi.

1.3. Terminu un pieņemtie apzīmējumi

Dokumentā izmantotie termini un apzīmējumi ir apkopoti Terminu un saīsinājumu indeksā [2].

1.4. Saistība ar citiem dokumentiem

Dokuments ir izstrādāts, balstoties uz šādiem dokumentiem:

- [1] „Valsts informācijas sistēmu savietotāja, Latvijas valsts portāla www.latvija.lv un elektronisko pakalpojumu izstrāde un uzturēšana”. Iepirkuma priekšmeta 3.daļa - VISS un portāla jaunu un esošo moduļu papildinājumu izstrāde, ieviešana, garantijas apkalpošana un uzturēšana saskaņā ar tehnisko specifikāciju. Terminu un saīsinājumu indekss. (VRAA-6_15_11_58-VISS_2010-TSI);
- [2] „Valsts informācijas sistēmu savietotāja, Latvijas valsts portāla www.latvija.lv un elektronisko pakalpojumu izstrāde un uzturēšana”. Iepirkuma priekšmeta 3.daļa - VISS un portāla jaunu un esošo moduļu papildinājumu izstrāde, ieviešana, garantijas apkalpošana un uzturēšana saskaņā ar tehnisko specifikāciju. VISS drošības talonu serviss. Programmētāja rokasgrāmata. (VRAA-6_15_11_58-VISS_2010-DTS-PR).
- [3] Valsts informācijas sistēmu savietotājs (VISS) un Vienotā valsts un pašvaldību pakalpojumu portāla www.latvija.lv pilnveidošana un uzturēšana. 3.daļa "VISS un Portāla jaunu un esošo moduļu papildinājumu izstrāde, ieviešana, garantijas apkalpošana un uzturēšana saskaņā

Dokumenta kods: VDAA-13_7_17_41 -VISS_2016-MM-II	Datums:08.10.2024	Versija:2.27
Datne: VISS_2016.II.MM_LVP3_v.2.27_tirraksts	Izstrādāja: M.Bambītis	Lpp.: 5 (35)
Valsts iepirkuma identifikācijas numurs VRAA/2010/10/ERAF/AK		

ar tehnisko specifikāciju". Maksājumu modulis un ledzīvotāju maksājumu nodrošināšana Programmatūras prasību specifikācija (VRAA-13_7_17_41-VISS_2016-MM-PPS).

- [4] Maksājumu modulis un iedzīvotāju maksājumu nodrošināšana: ārējās (VDAA-13_7_17_41-VISS_2016-MM_AS-PPA)
- [5] Maksājumu moduļa pielāgojumi neatkarīgai darbībai Lietotāja rokasgrāmata (VDAA/2023/57-LR-MM)

1.5. Dokumenta pārskats

Dokuments sastāv no šādiem nodaļumiem:

- Dokumenta ievads – aprakstīts dokumenta nolūks, termini un pieņemtie apzīmējumi, kā arī norādīta saistība ar citiem dokumentiem un materiāliem;
- Vispārīgs apraksts – sniedz ieskatu par risinājumu funkcionalitātes izmantošanu.

Dokumenta kods: VDAA-13_7_17_41-VISS_2016-MM-II	Datums:08.10.2024	Versija:2.27
Datne: VISS_2016.II.MM_LVP3_v.2.27_tīrraksts	Izstrādāja: M.Bambītis	Lpp.: 6 (35)
Valsts iepirkuma identifikācijas numurs VRAA/2010/10/ERAF/AK		

2. Izmantošanas scenāriji

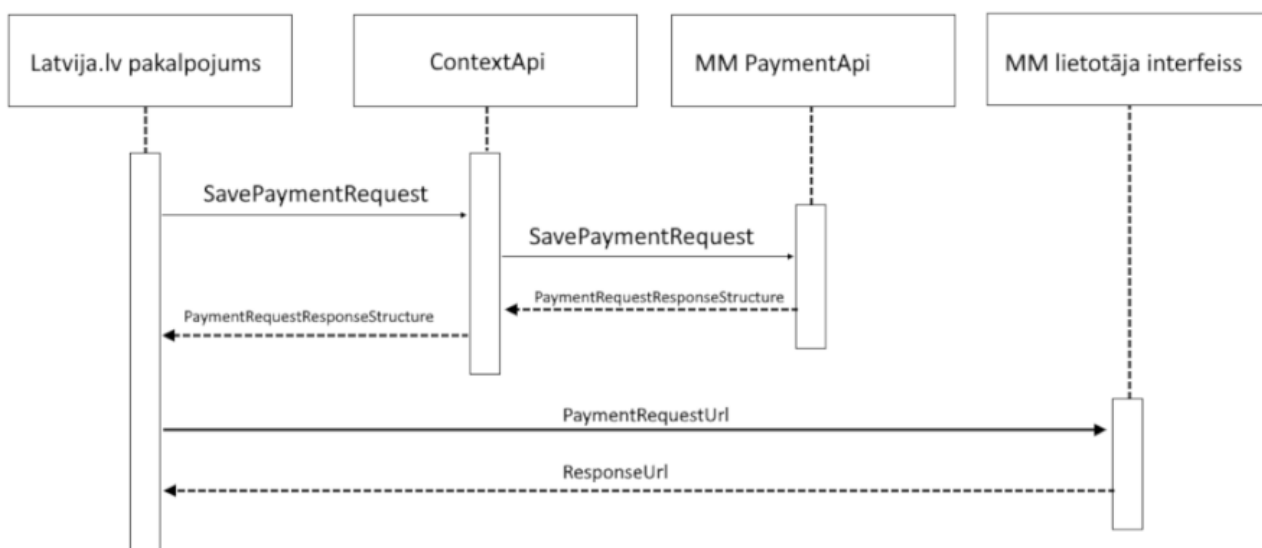
Iespējamie MM izmantošanas scenāriji

1. Portāla latvija.lv pakalpojumu apmaksas veikšana.
2. Ārēju pakalpojumu apmaksas veikšana.
3. Maksāšanas pieprasījuma statusa iegūšana.
4. Maksāšanas pieprasījuma attaisnojuma dokumenta datnes izgūšana PDF formātā.
5. Maksāšanas pieprasījuma attaisnojuma dokumenta datu izgūšana XML formātā.

2.1. Portāla latvija.lv pakalpojumu apmaksas veikšana

1. Pakalpojums caur epakalpojumu ietvara ContextApi izsauc MM PaymentApiservisu apmaksas procesa uzsākšanai un nodod PaymentsDataStructure objektu, kurš satur maksāšanas pieprasījumu. Tiek izveidots maksāšanas pieprasījums un atgriezts maksāšanas pieprasījuma identifikators un lietotāja interfeisa izsaukšanas hipersaite.
2. Pakalpojums izsauc atgrieztā MM lietotāja interfeisa hipersaiti. Hipersaitei iespējams pievienot 2.3 nodaļā minētos URL parametrus.
3. Pēc apmaksas veikšanas klients tiek atgriezts uz ResponseURL hipersaiti. Klients apmaksu var veikt līdz apmaksas termiņa iestāšanās laikam.
4. Pēc pēdējā maksājuma veiksmīgas apmaksas statusa saņemšanas no apmaksas nodrošinātāja, fona process nosūta klientam notifikāciju par maksāšanas pieprasījuma statusa maiņu, veicot GET pieprasījumu uz klienta norādīto ServerResponseUrl padodot tajā maksāšanas pieprasījuma unikālo guid. Ja ServerResponseURL netika padots, tad izsaukums netiek veikts.
5. Papildus maksāšanas pieprasījuma statusu un attaisnojuma dokumentu var pieprasīt jebkurā brīdī pēc izveidošanas.

Secību diagramma:



2.1.1. MM *izsaukumi,* *izmantojot* *Lvp.EservicePlatform.Backend.PaymentApi* mikroservisu

Mikroservisa API metožu izsaukšanai katram HTTP pieprasījumam body daļā JSON objektam jāsaturs :

- **MM_Cert** (klienta sertifikāta (skat.2.2. nodaļu) **publiskā** atslēga base64 kodējumā)

Visi datuma tipa parametri jānorāda ISO 8601 formātā ar laika joslu, piem.:

2019-12-31T15:45:00+3:00

Ja izsaukumā notikusi apstrādāta MM biznesa loģikas kļūda, tad atbildes header parametrā "MM-Error" tiek atgriezts kļūdas kods, atbilstoši Kļūdas teksts (ja tāds ir) tiek atgriezts MM-ErrorMessage header parametrā URL Encoded formātā.

2.1.1.1. Maksāšanas pieprasījuma izveidošana

IDENTIFIKATORS	LvpContext.Payments.SavePaymentRequest
APRAKSTS	Izveido jaunu maksāšanas pieprasījumu.

METODES IZSAUKŠANA

Adrese

POST /api/v1/payments/savepaymentrequest

HEADER parametri

Content-Type - application/json

PATH parametri

Nav.

QUERY parametri

Nav.

BODY parametri

PaymentsDataStructure JSON formātā atbilstoši MM specifikācijai (VDAA-13_7_17_41-VISS_2016-MM-AS-PPA) [3]. Par cenrāža pozīciju un veidņu (PriceList un ServiceStep) vērtībām skatīt MM portāla lietotāja rokasgrāmatu (VDAA/2023/57-LR-MM)[5]

Piemērs

Dokumenta kods: VDAA-13_7_17_41-VISS_2016-MM-II	Datums:08.10.2024	Versija:2.27
Datne: VISS_2016.II.MM_LVP3_v.2.27_tīrāksts	Izstrādāja: M.Bambītis	Lpp.: 8 (35)
Valsts iepirkuma identifikācijas numurs VRAA/2010/10/ERAF/AK		

POST /api/v1/payments/savepaymentrequest HTTP/1.1

```
{
  "MM_Cert": "MIIM1QIBAzCCDJEGCSqGSIB3DQEHAaCCDIIEggx+MII ... P63AgIH0A==",
  "PaymentsDataStructure": {
    "PaymentRequest": {
      "AllowedBanks": [
        {"SwiftCode": "DEMO0000"},
        {"SwiftCode": "FIRSTDATA"},
        {"SwiftCode": "HABALV22"}
      ],
      "ClientInfo": {
        "ClientAddress": "Rātslaukums 1, LV-1050, Rīga",
        "ClientCode": "07018710513",
        "ClientEMail": "e@p.lv",
        "ClientFirstName": "Ernests",
        "ClientName": "Paraudziņš",
        "ClientPhone": "26620000"
      },
      "ExpiryDate": "2020-09-30T14:00:00+03:00",
      "NoInvoice": true,
      "PaymentObject": "DEM11 payment",
      "PriceList": [{
        "Item": 12.00,
        "ItemElementName": "Quantity",
        "Position": "2346"
      }],
      "ResponseUrl": "https://tv-vraa.zzdati.lv/MMTests/S2S/?id=",
      "ServerResponseUrl": "https://tv-vraa.zzdati.lv/MMTests/S2S/ok1?id=",
      "ServiceStep": "03G10",
      "SwiftCode": "DEMO0000",
      "ReturnRedirectUrl": "https://latvija.lv/EP21/"
    }
  }
}
```

IZVADDATI

PaymentRequestResponseStructure JSON objekts (atbilstoši [3] 2.1.1 nodaļai), ar laukiem:

- Maksāšanas pieprasījuma identifikators
- Maksājumu moduļa izsaukšanas URL. Hipersaitei atļauts pievienot 2.1 nodaļas 2. punktā minētos URL parametrus

Kļūdas

- HTTP 500 – kļūda pieprasījuma apstrādē
- HTTP 200 – maksāšanas pieprasījums veiksmīgi saglabāts.

Piemērs

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
  "PaymentRequestId": "05ae69d8-2be1-4a63-ae93-9d0e42e254f1",
  "paymentRequestUrl": https://epakvisstv.vraa.gov.lv/MM/WebUI2/Payments/?id=05ae69d8-2be1-4a63-ae93-9d0e42e254f1
}
```

2.1.1.2. Maksāšanas pieprasījuma statusa pārbaude

IDENTIFIKATORS	LvpContext.Payments.CheckPaymentRequestStatus
APRAKSTS	Atgriež maksāšanas pieprasījuma statusu

METODES IZSAUKŠANA

Adrese

POST /api/v1/payments/checkpaymentrequeststatus

HEADER parametri

Content-Type – application/json

PATH parametri

Nav.

QUERY parametri

Nav

BODY parametri

PaymentRequestId – maksāšanas pieprasījuma unikālais identifikators.

Piemērs

```
POST /api/v1/payments/checkpaymentrequeststatus HTTP/1.1

{
  "MM_Cert": "MIIM1QIBAzCCDJEGCSqGSIB3DQEHAaCCDIEggx+MII ... P63AgIH0A==",
  "PaymentRequestId": "a2a4b66a-a0b7-4b1d-bcc3-3d5a8d30b2b1"
}
```

IZVADDATI

Maksāšanas pieprasījuma statusa kods

Kļūdas

- HTTP 500 – kļūda pieprasījuma apstrādē
- HTTP 404 – maksāšanas pieprasījums ar norādīto Id netika atrasts
- HTTP 200 – maksāšanas pieprasījuma statuss veiksmīgi atgriezts.

Piemērs

```
HTTP/1.1 200 OK
Content-Type: text/plain charset=utf-8

E
```

2.1.1.3. Maksāšanas pieprasījumu izgūšana

IDENTIFIKATORS	LvpContext.Payments.GetPaymentRequests
APRAKSTS	Atgriež maksāšanas pieprasījumu sarakstu

METODES IZSAUKŠANA

Adrese

POST /api/v1/payments/getpaymentrequests

Dokumenta kods: VDAA-13_7_17_41 -VISS_2016-MM-II	Datums:08.10.2024	Versija:2.27
Datne: VISS_2016.II.MM_LVP3_v.2.27_tīrāksts	Izstrādāja: M.Bambītis	Lpp.: 10 (35)
Valsts iepirkuma identifikācijas numurs VRAA/2010/10/ERAF/AK		

HEADER parametri

Content-Type – application/json

PATH parametri

Nav.

QUERY parametri

Nav

BODY parametri

CreatedDateFrom – maksāšanas pieprasījuma izrakstīšanas datums no.

CreatedDateTill – maksāšanas pieprasījuma izrakstīšanas datums līdz

PaymentDateFrom – apmaksas datums no

PaymentDateTill – apmaksas datums līdz

ExpirationDateFrom – pieprasījuma derīguma termiņa datums no

ExpirationDateTill – pieprasījuma derīguma termiņa datums līdz

AmountFrom – maksājuma summa no

AmountTill – maksājuma summa līdz

OnlyNotPaid – tikai neapmaksātos pieprasījumus

Currency – maksājuma valūta

Piemērs

```
POST /api/v1/payments/getpaymentrequests HTTP/1.1

{
  "MM_Cert": "MIIM1QIBAzCCDJEGCSqGSIB3DQEHAaCCDIIIEggx+MII ... P63AgIH0A==",
  "CreatedDateFrom": "2020-01-31T12:41:00+3:00",
  "CreatedDateTill": "2020-08-14T12:41:00+3:00",
  "AmountFrom": "7.99",
  "OnlyNotPaid": false
}
```

IZVADDATI

Maksāšanas pieprasījumu saraksts JSON formātā body daļā (atbilstoši [3] 2.4.1 nodaļai).

Kļūdas

- HTTP 500 – kļūda pieprasījuma apstrādē
- HTTP 200 – maksāšanas pieprasījumu saraksts veiksmīgi atgriezts.

Piemērs

Dokumenta kods: VDAA-13_7_17_41 -VISS_2016-MM-II	Datums: 08.10.2024	Versija: 2.27
Datne: VISS_2016.II.MM_LVP3_v.2.27_tiraksts	Izstrādāja: M.Bambītis	Lpp.: 11 (35)
Valsts iepirkuma identifikācijas numurs VRAA/2010/10/ERAF/AK		

```

HTTP/1.1 200 OK
Content-Type: application/json charset=utf-8

[
  {
    "PaymentRequestUniqueId": "a2a4b66a-a0b7-4b1d-bcc3-3d5a8d30b2b1",
    "Number": "ABC-123",
    "StartDate": "2020-07-24",
    "ExpirationDate": "2020-08-01",
    "ServiceProviderName": "VDAA",
    "ServiceName": "Pakalpojuma nosaukums",
    "TotalAmount": 123.45,
    "Currency": "EUR",
    "UsageStatus": "Izmantots"
  },
  {
    "PaymentRequestUniqueId": "b6a4b66a-b1b7-4b1d-bcc3-3d5a8d30b3c2",
    "Number": "ABC-456",
    "StartDate": "2020-07-24",
    "ExpirationDate": "2020-08-01",
    "ServiceProviderName": "VDAA",
    "ServiceName": "Pakalpojuma nosaukums 2",
    "TotalAmount": 8.20,
    "Currency": "EUR",
    "UsageStatus": "Izmantots"
  }
]

```

2.1.1.4. Maksājumu izgūšana

IDENTIFIKATORS	LvpContext.Payments.GetPayments
APRAKSTS	Atgriež maksājumu sarakstu

METODES IZSAUKŠANA

Adrese

POST /api/v1/payments/getpayments

HEADER parametri

Content-Type - application/json

PATH parametri

Nav.

QUERY parametri

Nav

BODY parametri

PaymentDateFrom – apmaksas datums no.

PaymentDateTill - apmaksas datums līdz

AmountFrom – maksājuma summa no

AmountTill – maksājuma summa līdz

Currency – maksājuma valūta

Piemērs

Dokumenta kods: VDAA-13_7_17_41 -VISS_2016-MM-II	Datums: 08.10.2024	Versija: 2.27
Datne: VISS_2016.II.MM_LVP3_v.2.27_tirraksts	Izstrādāja: M.Bambītis	Lpp.: 12 (35)
Valsts iepirkuma identifikācijas numurs VRAA/2010/10/ERAF/AK		

POST /api/v1/payments/getpayments HTTP/1.1

```
{
  "MM_Cert": "MIIM1QIBAzCCDJEGCSqGSib3DQEHAaCCDIIEggx+MII ... P63AgIH0A==",
  "PaymentDateFrom": "2019-12-31T12:41:00+3:00",
  "PaymentDateTill": "2020-09-22T00:00:00+3:00",
  "AmountFrom": "7.99",
  "Currency": "EUR"
}
```

IZVADDATI

Maksājumu saraksts JSON formātā (atbilstoši [3] 2.4.2 nodaļai).

Kļūdas

- HTTP 500 – kļūda pieprasījuma apstrādē
- HTTP 200 – maksājumu saraksts veiksmīgi atgriezts.

Piemērs

```
HTTP/1.1 200 OK
Content-Type: application/json charset=utf-8

[
  {
    "PaymentRequestUniqueId": "a2a4b66a-a0b7-4b1d-bcc3-3d5a8d30b2b1",
    "PaymentDate": "2020-07-24",
    "ExternalId": "2589",
    "ServiceName": "Pakalpojums 1",
    "PayeeAccountNumber": "LV48AHAB0551014343801",
    "BankInfo": "Maksājuma mērķis 1",
    "Amount": 123.45,
    "Currency": "EUR",
    "StatusCode": "R"
  },
  {
    "PaymentRequestUniqueId": "a2a4b66a-a0b7-4b1d-bcc3-3d5a8d30b2b1",
    "PaymentDate": "2020-07-24",
    "ExternalId": "11258",
    "ServiceName": "Pakalpojums 2",
    "PayeeAccountNumber": "LV48AHAB0551014343801",
    "BankInfo": "Maksājuma mērķis 2",
    "Amount": 42.69,
    "Currency": "EUR",
    "StatusCode": "E"
  }
]
```

2.1.1.5. Maksājuma statusa pārbaude

IDENTIFIKATORS	LvpContext.Payments.CheckPaymentStatus
APRAKSTS	Maksājumu statusu pārbaude pa pozīcijām no ārējiem pakalpojumiem

METODES IZSAUKŠANA

Adrese

POST /api/v1/payments/checkpaymentstatus

HEADER parametri

Dokumenta kods: VDAA-13_7_17_41 -VISS_2016-MM-II	Datums: 08.10.2024	Versija: 2.27
Datne: VISS_2016.II.MM_LVP3_v.2.27_tīrāksts	Izstrādāja: M.Bambītis	Lpp.: 13 (35)
Valsts iepirkuma identifikācijas numurs VRAA/2010/10/ERAF/AK		

Content-Type - application/json

PATH parametri

Nav.

QUERY parametri

Nav

BODY parametri

PaymentRequestId – maksāšanas pieprasījuma unikālais identifikators.

Position – cenrāža pozīcijas kods

Piemērs

```
POST /api/v1/payments/checkpaymentstatus HTTP/1.1

{
  "MM_Cert": "MIIM1QIBAzCCDJECSqGSIB3DQEHAaCCDIEggx+MII ... P63AgIH0A==",
  "paymentRequestId": "a2a4b66a-a0b7-4b1d-bcc3-3d5a8d30b2b1",
  "position": "172"
}
```

IZVADDATI

PaymentStatusResponseStructure JSON objekts (atbilstoši [3] 2.2.3 nodaļai), ar laukiem:

- Maksājuma statusa kods
- Maksāšanas sniedzēja SWIFT kods

Kļūdas

- HTTP 500 – kļūda pieprasījuma apstrādē
- HTTP 404 – maksāšanas pieprasījums ar norādīto Id netika atrasts
- HTTP 200 – maksājuma statuss veiksmīgi atgriezts.

Piemērs

```
HTTP/1.1 200 OK
Content-Type: application/json charset=utf-8

{
  "RequestStatus": "R"
  "PaymentProvider": "LVHABA22"
}
```

2.1.1.6. Maksāšanas pieprasījumu statusu pārbaude un apmaksas veida izgūšana

IDENTIFIKATORS	LvpContext.Payments.CheckPaymentRequestStatusWpt
APRAKSTS	Maksāšanas pieprasījumu statusu pārbaude un apmaksas veida izgūšana no ārējiem pakalpojumiem

METODES IZSAUKŠANA

Adrese

POST /api/v1/paymentrequest/checkpaymentrequeststatuswpt

HEADER parametri

Dokumenta kods: VDAA-13_7_17_41 -VISS_2016-MM-II	Datums:08.10.2024	Versija:2.27
Datne: VISS_2016.II.MM_LVP3_v.2.27_tīrāksts	Izstrādāja: M.Bambītis	Lpp.: 14 (35)
Valsts iepirkuma identifikācijas numurs VRAA/2010/10/ERAF/AK		

Content-Type - application/json

PATH parametri

Nav.

QUERY parametri

Nav

BODY parametri

PaymentRequestId – maksāšanas pieprasījuma unikālais identifikators.

Piemērs

```
POST /api/v1/payments/checkpaymentrequeststatuswpt HTTP/1.1

{
  "MM_Cert": "MIIM1QIBAzCCDJEGCSqGSIB3DQEHAAACCDIIEggx+MII ... P63AgIH0A==",
  "paymentRequestId": "a2a4b66a-a0b7-4b1d-bcc3-3d5a8d30b2b1"
}
```

IZVADDATI

PaymentRequestStatusResponseStructure JSON objekts (atbilstoši [3] 2.2.4 nodaļai) ar laukiem

- Maksājuma statusa kods
- Maksāšanas sniedzēja(-u) SWIFT kods

Kļūdas

- HTTP 500 – kļūda pieprasījuma apstrādē
- HTTP 404 – maksāšanas pieprasījums ar norādīto Id netika atrasts
- HTTP 200 – maksājuma statuss veiksmīgi atgriezts.

Piemērs

```
HTTP/1.1 200 OK
Content-Type: application/json charset=utf-8

{
  "RequestStatus": "R"
  "PaymentProviders": "LVHABA22;FIRSTDATA"
}
```

2.1.1.7. Rēķina izveidošana

IDENTIFIKATORS	LvpContext.Payments.CreateInvoice
APRAKSTS	Izveido rēķina PDF.

METODES IZSAUKŠANA

Adrese

POST /api/v1/payments/createinvoice

HEADER parametri

Content-Type - application/json

PATH parametri

Nav.

Dokumenta kods: VDAA-13_7_17_41-VISS_2016-MM-II	Datums:08.10.2024	Versija:2.27
Datne: VISS_2016.II.MM_LVP3_v.2.27_tirraksts	Izstrādāja: M.Bambītis	Lpp.: 15 (35)
Valsts iepirkuma identifikācijas numurs VRAA/2010/10/ERAF/AK		

QUERY parametri

Nav.

BODY parametri

Id – maksāšanas pieprasījuma unikālais identifikators

Name – klienta vārds/nosaukums

Code – klienta personas kods/reģ. Nr.

Address – klienta adrese

Piemērs

```
POST /api/v1/payments/createinvoice HTTP/1.1

{
  "MM_Cert": "MIIM1QIBAzCCDJEGCSqGSIB3DQEHAaCCDIEggx+MII ... P63AgIH0A==",
  "Id": "a2a4b66a-a0b7-4b1d-bcc3-3d5a8d30b2b1",
  "Name": "SIA ZZ Dats",
  "Code": "40003278467",
  "Address": "Elizabetes 41/43, Rīga"
}
```

IZVADDATI

MM DB izveidotā Repository tabulas ieraksta Id

Kļūdas

- HTTP 500 – kļūda pieprasījuma apstrādē
- HTTP 404 – maksāšanas pieprasījums ar norādīto Id netika atrasts
- HTTP 200 – rēķins veiksmīgi izveidots.

Piemērs

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
  "id": 12128
}
```

Pēc veiksmīga izsaukuma rēķina datni iespējams izgūt, izmantojot MM REST (2.4. nodaļa) un SOAP (2.6. nodaļa) servisu.

2.1.1.8. Maksāšanas pieprasījuma atcelšana

IDENTIFIKATORS	LvpContext.Payments.CancelPaymentRequest
APRAKSTS	Uzstāda maksāšanas pieprasījumam statusu “Manuāli atcelts”.

METODES IZSAUKŠANA**Adrese**

POST /api/v1/payments/cancelpaymentrequest

Dokumenta kods: VDAA-13_7_17_41 -VISS_2016-MM-II	Datums: 08.10.2024	Versija: 2.27
Datne: VISS_2016.II.MM_LVP3_v.2.27_tiraksts	Izstrādāja: M.Bambītis	Lpp.: 16 (35)
Valsts iepirkuma identifikācijas numurs VRAA/2010/10/ERAF/AK		

HEADER parametri

Content-Type – application/json

PATH parametri

Nav.

QUERY parametri

Nav.

BODY parametri

PaymentRequestId – maksāšanas pieprasījuma unikālais identifikators

Piemērs

```
POST /api/v1/payments/cancelpaymentrequest HTTP/1.1

{
  "MM_Cert": "MIIM1QIBAzCCDJECSqGSIB3DQEHAAcCDIIEgqx+MII ... P63AgIH0A==",
  "PaymentRequestId" : "a2a4b66a-a0b7-4b1d-bcc3-3d5a8d30b2b1"
}
```

IZVADDATI

Nav

Kļūdas

- HTTP 500 – kļūda pieprasījuma apstrādē
- HTTP 404 – maksāšanas pieprasījums ar norādīto Id netika atrasts
- HTTP 200 – maksāšanas pieprasījums veiksmīgi atcelts.

Piemērs

```
HTTP/1.1 200 OK
```

2.2. Ārēju pakalpojumu apmaksas veikšana

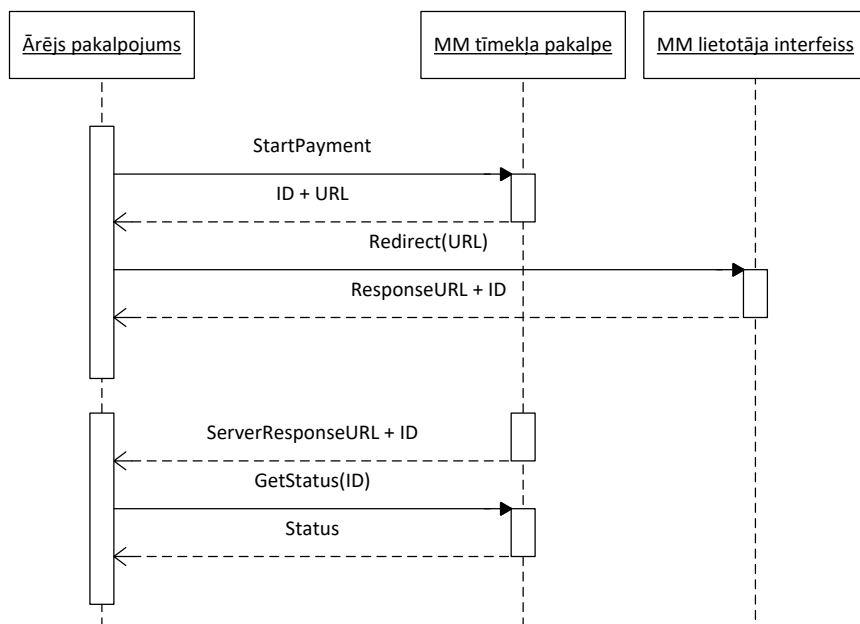
1. Pieslēdzoties tiek pārbaudīts klienta sertifikāts. Sertifikātu izsniedz pakalpojuma turētājs – VDAA. Sertifikātam jābūt uzstādītam direktorijā Local Machine\Personal\Certificates. Jābūt uzstādītiem arī saknes sertifikātiem, lai klienta sertifikāts validētos. Saknes sertifikātiem jābūt uzstādītiem direktorijā Local Machine\Trusted Root Certification Authorities\Certificates;
2. Pakalpojumi izsauc tīmekļa pakalpi apmaksas procesa uzsākšanai un izveido maksāšanas pieprasījumu. Pakalpojuma apmaksai jābūt definētam cenrādim, maksājuma veidnei un maksāšanas pozīcijām atbilstoši instrukcijām [5] dokumentā. Tiek atgriezts maksāšanas pieprasījuma identifikators un lietotāja interfeisa izsaukšanas hipersaite.
3. Pakalpojumi izsauc atgrieztā MM lietotāja interfeisa hipersaiti. Hipersaitei atļauts pievienot 2.3 nodaļā minētos URL parametrus.
4. Pēc apmaksas veikšanas klients tiek atgriezts uz ResponseURL hipersaiti. Klients apmaksu var veikt līdz apmaksas termiņa iestāšanās laikam.
5. Pēc pēdējā maksājuma statusa iegūšanas no apmaksas nodrošinātāja vai pēc apmaksas termiņa iestāšanās, neapmaksāta maksāšanas pieprasījuma gadījumā, tiek izsaukta

ServerResponseURL parametrā padotā hipersaite paziņojuma par maksāšanas pieprasījuma statusa maiņu nodošanai. Ja ServerResponseURL netika padots, tad izsaukums netiek veikts.

6. Pēc paziņojuma par maksāšanas pieprasījuma statusa maiņu saņemšanas pakalpojumam jāizsauc maksāšanas pieprasījuma statusa saņemšana maksāšanas pieprasījuma statusa noskaidrošanai.
7. Maksāšanas pieprasījuma statusu un attaisnojuma dokumentu var pieprasīt jebkurā brīdī pēc izveidošanas.

2.2.1. Maksāšanas pieprasījuma izveidošana

Secību diagramma:



.NET izmantošanas koda piemērs:

```

PaymentRequestServiceReference.PaymentRequestServiceClient client =
    new
PaymentRequestServiceReference.PaymentRequestServiceClient();
    string error = null;
Client.ClientCredentials.ClientCertificate.SetCertificate(
    System.Security.Cryptography.X509Certificates.StoreLocation.CurrentUser,
        System.Security.Cryptography.X509Certificates.StoreName.My,
    System.Security.Cryptography.X509Certificates.X509FindType.FindByThumbprint,
        "THUMBPRINT");

try
{
    var request = new PaymentsDataStructure()
    {
        PaymentRequest = new PaymentRequestStructure()
        {
            //Ja nav norādīta neviena AllowedBankStructure - tad tiek
            attēlotas visas apmaksāšanas iespējas.
            AllowedBanks = new AllowedBankStructure[]
            {

```

```

        new AllowedBankStructure() { SwiftCode = "DEMO0000" },
        new AllowedBankStructure() { SwiftCode = "HABALV22" }
    },
    ClientInfo = new ClientInfoStructure()
    {
        ClientCode = "31129900000",
        ClientFirstName = "Ernests",
        ClientName = "Paraudziņš",
        ClientEMail = "e@p.lv",
        ClientPhone = "26620000",
        ClientAddress = "Rātslaukums 1, LV-1050, Rīga"
    },
    ExpiryDate = DateTime.Now.AddDays(5),
    InvoiceIssueDate = DateTime.Now.AddDays(10),
    NoInvoice = true, //vai nepieciešams attainsojuma dokuments
    PaymentObject = "PmtInfo",
    PriceList = new PriceListStructure[]
    {
        new PriceListStructure()
        {
            Item = 4.56m,
            Position = "1",
            ItemElementName = ItemChoiceType.Quantity,
        },
        new PriceListStructure()
        {
            Item = 0.54m,
            Position = "2",
            ItemElementName = ItemChoiceType.Amount,
        }
    },
    ResponseUrl = "http://jusu-response-url",
    ServerResponseUrl = "http://jusu-response-url-s2s",
    ReturnRedirectUrl = "https://latvija.lv/EP21/",
    ServiceStep = "DEM11",
    SwiftCode = "DEMO0000", //Bankas SWIFT kods, ar kuru
    autorizējies lietotājs
    }
};

PaymentRequestResponseStructure result =
client.SavePaymentRequest(request);

//saglabāt identifikatoru
var paymentRequestId = result.PaymentRequestId;

//izsaukt lietotāja interfeisu
RedirectTo(result.PaymentRequestUrl);
}
catch (FaultException<PaymentRequestFault> ex)
{
    error = ex.Detail.Message;
    client.Abort();
}
}

```

PHP izmantošanas koda piemērs:

```

<?php
header('Content-Type: text/html; charset=utf-8');

$wsdlUrl = 'https://tv-
vraa.zzdati.lv/services/VRAA.VISS2.MM.Endpoint/v1.0/WCFX509/PaymentRequestService.svc?singleWsdl';
// Sertifikātu no pfx uz pem konvertē ar komandu:

```

Dokumenta kods: VDAA-13_7_17_41-VISS_2016-MM-II	Datums:08.10.2024	Versija:2.27
Datne: VISS_2016.II.MM_LVP3_v.2.27_tiraksts	Izstrādāja: M.Bambītis	Lpp.: 19 (35)
Valsts iepirkuma identifikācijas numurs VRAA/2010/10/ERAF/AK		

```
// openssl pkcs12 -in fails.pfx -out fails.pem
$authCertPath = '/path/to/mm.pem';
$authCertPass = 'demo';

$client = new SoapClient($wsdlUrl, array('soap_version' => SOAP_1_1,
'local_cert' => $authCertPath, 'passphrase' => $authCertPass));

try
{
    $resp = $client->SavePaymentRequest(array(
        'paymentRequest' => array
        (
            'Payment' => null,
            'PaymentRequest' => array
            (
                'ClientInfo' => array
                (
                    'ClientCode' =>
'31129900000',
                    'ClientFirstName' =>
'Ernests',
                    'ClientName' =>
'Paraudziņš',
                    'ClientEMail' => 'e@p.lv',
                    'ClientPhone' =>
'26620000',
                    'ClientAddress' =>
'Rātslaukums 1, LV-1050, Rīga'
                ),
                'ParentPaymentRequestID' => null,
                'PaymentObject' => '02Y70
payment',
                'PriceList' => array(
                    array('Item' => 1,
'ItemElementName' => 'Quantity', 'Position' => 53),
                    array('Item' => 12.12,
'ItemElementName' => 'Amount', 'Position' => 54),
                ),
                'ResponseUrl' => 'http://jusu-
response-url',
                'ReturnRedirectUrl' =>
'http://jusu-pakalpojuma-url',
                'ServerResponseUrl' =>
'http://jusu-response-url-s2s',
                'ServiceStep' => '02Y70',
                'SwiftCode' => 'DEMO0000',
            ),
            'PaymentRequestPPK' => null,
            'PaymentRequestResponse' => null,
            'PaymentRequestStatus' => null
        )
    ));

    // Saglabā identifikatoru
    $paymentRequestId = $resp->SavePaymentRequestResult->PaymentRequestId;

    // Izsauc lietotāja saskarni
    header('Location: ' . $resp->SavePaymentRequestResult-
>PaymentRequestUrl);
}
catch(SoapFault $ex)
{
    // TODO: Nomainīt uz kļūdu apstrādi
}
```

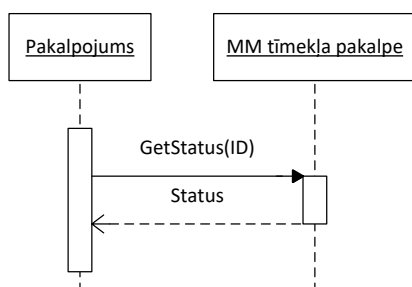
```

        die('SoapFault: ' . $ex->getMessage());
    }

```

2.2.2. Maksāšanas pieprasījuma statusa pārbaude

Secību diagramma:



.NET izmantošanas koda piemērs:

```

PaymentRequestServiceReference.PaymentRequestServiceClient client =
    new
PaymentRequestServiceReference.PaymentRequestServiceClient();
string error = null;

try
{
    //Izmantojam maksāšanas pieprasījuma identifikatoru no
SavePaymentRequest()
    string result = client.CheckPaymentRequestStatus
(paymentRequestId);
    //result = Maksāšanas pieprasījuma statuss (P = Procesā, E =
Izpildīts, R = ExpireDate, O = Other, iespējams tikai ja apmaksas nodrošinātājs
atgriež)
}
catch (FaultException<PaymentRequestFault> ex)
{
    error = ex.Detail.Message;
    client.Abort();
}

```

PHP izmantošanas koda piemērs:

```

<?php
header('Content-Type: text/html; charset=utf-8');

$wsdlUrl = 'https://tv-
vraa.zzdati.lv/services/VRAA.VISS2.MM.Endpoint/v1.0/WCFX509/PaymentRequestServic
e.svc?singleWsdl';
// Sertifikātu no pfx uz pem konvertē ar komandu:
// openssl pkcs12 -in fails.pfx -out fails.pem
$authCertPath = '/path/to/mm.pem';
$authCertPass = 'demo';

```

Dokumenta kods: VDAA-13_7_17_41 -VISS_2016-MM-II	Datums:08.10.2024	Versija:2.27
Datne: VISS_2016.II.MM_LVP3_v.2.27_tīrāksts	Izstrādāja: M.Bambītis	Lpp.: 21 (35)
Valsts iepirkuma identifikācijas numurs VRAA/2010/10/ERAF/AK		

```
$client = new SoapClient($wsdlUrl, array('soap_version' => SOAP_1_1,
'local_cert' => $authCertPath, 'passphrase' => $authCertPass));

$paymentRequestId = '99e083d2-688f-481a-bb85-420eb11e6927';

try
{
    $resp = $client->CheckPaymentRequestStatus(array(
        'paymentRequestId' => $paymentRequestId
    ));

    // TODO: Darbības, ja pieprasījums izdevies veiksmīgi
    // Maksāšanas pieprasījuma statuss (P = Procesā, E = Izpildīts, R =
    ExpireDate, O = Other, iespējams tikai ja apmaksas nodrošinātājs atgriež)
    echo 'Statuss: ' . $resp->CheckPaymentRequestStatusResult . '<br />';
}
catch(SoapFault $ex)
{
    // TODO: Nomainīt uz kļūdu apstrādi
    die('SoapFault: ' . $ex->getMessage());
}
```

2.2.3. Maksāšanas pieprasījumu izgūšana

.NET izmantošanas koda piemērs:

```
PaymentRequestServiceReference.PaymentRequestServiceClient client =
    new
PaymentRequestServiceReference.PaymentRequestServiceClient();
string error = null;

try
{
var position="123";
    //Izmantojam maksāšanas pieprasījuma identifikatoru no
SavePaymentRequest()
    var result = client.GetPaymentRequests (
DateTime.Now.AddDays(-3), DateTime.Now, null, null, null, null, 10, 100, false,
"EUR");

}
catch (FaultException<PaymentRequestFault> ex)
{
    error = ex.Detail.Message;
    client.Abort();
}
```

PHP izmantošanas koda piemērs:

```
<?php
header('Content-Type: text/html; charset=utf-8');

$wsdlUrl = 'https://tv-
vraa.zzdati.lv/services/VRAA.VISS2.MM.Endpoint/v1.0/WCFX509/PaymentRequestService.svc?singleWsdl';
// Sertifikātu no pfx uz pem konvertē ar komandu:
// openssl pkcs12 -in fails.pfx -out fails.pem
$authCertPath = '/path/to/mm.pem';
$authCertPass = 'demo';
```

Dokumenta kods: VDAA-13_7_17_41-VISS_2016-MM-II	Datums:08.10.2024	Versija:2.27
Datne: VISS_2016.II.MM_LVP3_v.2.27_tiraksts	Izstrādāja: M.Bambītis	Lpp.: 22 (35)
Valsts iepirkuma identifikācijas numurs VRAA/2010/10/ERAF/AK		

```

$client = new SoapClient($wsdlUrl, array('soap_version' => SOAP_1_1,
'local_cert' => $authCertPath, 'passphrase' => $authCertPass));

$paymentRequestId = '99e083d2-688f-481a-bb85-420eb11e6927';
$position = '123';

try
{
    $resp = $client->GetPaymentRequests(array(
        'createdDateFrom' => getdate(),
        'createdDateTill' => getdate(),
        'paymentDateFrom' => null,
        'paymentDateTill' => null,
        'expirationDateFrom' => null,
        'expirationDateTill' => null,
        'amountFrom' => 10,
        'amountTill' => 100,
        'onlyNotPaid' => false,
        'currency' => "EUR"

    ));

    echo 'Statuss: ' . $resp-> GetPaymentRequestsResult . '<br />';
}
catch(SoapFault $ex)
{
    // TODO: Nomainīt uz kļūdu apstrādi
    die('SoapFault: ' . $ex->getMessage());
}

```

2.2.4. Maksājumu izgūšana

.NET izmantošanas koda piemērs:

```

PaymentRequestServiceReference.PaymentRequestServiceClient client =
    new
PaymentRequestServiceReference.PaymentRequestServiceClient();
string error = null;

try
{
    var result = client.GetPayments (
DateTime.Now.AddDays(-3), DateTime.Now, 10, 100, "EUR");

}
catch (FaultException<PaymentRequestFault> ex)
{
    error = ex.Detail.Message;
    client.Abort();
}

```

PHP izmantošanas koda piemērs:

```

<?php
header('Content-Type: text/html; charset=utf-8');

```

Dokumenta kods: VDAA-13_7_17_41-VISS_2016-MM-II	Datums: 08.10.2024	Versija: 2.27
Datne: VISS_2016.II.MM_LVP3_v.2.27_tīrāksts	Izstrādāja: M.Bambītis	Lpp.: 23 (35)
Valsts iepirkuma identifikācijas numurs VRAA/2010/10/ERAF/AK		

```

$wsdlUrl = 'https://tv-
vraa.zzdati.lv/services/VRAA.VISS2.MM.Endpoint/v1.0/WCFX509/PaymentRequestService.svc?singleWsdl';
// Sertifikātu no pfx uz pem konvertē ar komandu:
// openssl pkcs12 -in fails.pfx -out fails.pem
$authCertPath = '/path/to/mm.pem';
$authCertPass = 'demo';

$client = new SoapClient($wsdlUrl, array('soap_version' => SOAP_1_1,
'local_cert' => $authCertPath, 'passphrase' => $authCertPass));

try
{
    $resp = $client->GetPayments(array(
        'paymentDateFrom' => getdate(),
        'paymentDateTill' => getdate(),
        'amountFrom' => 10,
        'amountTill' => 100,
        'currency' => "EUR"

    ));

    echo 'Statuss: ' . $resp->GetPaymentsResult . '<br />';
}
catch(SoapFault $ex)
{
    // TODO: Nomainīt uz kļūdu apstrādi
    die('SoapFault: ' . $ex->getMessage());
}

```

2.2.5. Maksājuma statusa pārbaude

.NET izmantošanas koda piemērs:

```

PaymentRequestServiceReference.PaymentRequestServiceClient client =
    new
PaymentRequestServiceReference.PaymentRequestServiceClient();
string error = null;

try
{
var position="123";
    //Izmantojam maksāšanas pieprasījuma identifikatoru no
SavePaymentRequest()
    string result = client.CheckPaymentStatus (paymentRequestId,
position);
    //result = Maksājuma statuss (P = Procesā, E = Izpildīts, R =
ExpireDate, O = Other, iespējams tikai ja apmaksas nodrošinātājs atgriež)
}
catch (FaultException<PaymentRequestFault> ex)
{
    error = ex.Detail.Message;
    client.Abort();
}

```

PHP izmantošanas koda piemērs:

Dokumenta kods: VDAA-13_7_17_41-VISS_2016-MM-II	Datums:08.10.2024	Versija:2.27
Datne: VISS_2016.II.MM_LVP3_v.2.27_tiraksts	Izstrādāja: M.Bambītis	Lpp.: 24 (35)
Valsts iepirkuma identifikācijas numurs VRAA/2010/10/ERAF/AK		


```
<?php
header('Content-Type: text/html; charset=utf-8');

$wsdlUrl = 'https://tv-
vraa.zzdati.lv/services/VRAA.VISS2.MM.Endpoint/v1.0/WCFX509/PaymentRequestService.svc?singleWsdl';
// Sertifikātu no pfx uz pem konvertē ar komandu:
// openssl pkcs12 -in fails.pfx -out fails.pem
$authCertPath = '/path/to/mm.pem';
$authCertPass = 'demo';

$client = new SoapClient($wsdlUrl, array('soap_version' => SOAP_1_1,
'local_cert' => $authCertPath, 'passphrase' => $authCertPass));

$paymentRequestId = '99e083d2-688f-481a-bb85-420eb11e6927';
$position = '123';

try
{
    $resp = $client->CheckPaymentStatus(array(
        'paymentRequestId' => $paymentRequestId,
        'position' => $position
    ));

    // Maksāsājuma statuss (P = Procesā, E = Izpildīts, R = ExpireDate, O =
    // Other, iespējams tikai ja apmaksas nodrošinātājs atgriež)
    echo 'Statuss: ' . $resp->CheckPaymentStatusResult . '<br />';
}
catch(SoapFault $ex)
{
    // TODO: Nomainīt uz kļūdu apstrādi
    die('SoapFault: ' . $ex->getMessage());
}
```

2.2.6. Rēķina izveidošana

.NET izmantošanas koda piemērs:

```
PaymentRequestServiceReference.PaymentRequestServiceClient client =
    new
    PaymentRequestServiceReference.PaymentRequestServiceClient();
string error = null;
Client.ClientCredentials.ClientCertificate.SetCertificate(
    System.Security.Cryptography.X509Certificates.StoreLocation.LocalMachine,
    System.Security.Cryptography.X509Certificates.StoreName.My,

    var name = "Ernestis Paraudziņš";
    var code = "31129910101";
    var address = "Elizabetes 41/43, Rīga";

    System.Security.Cryptography.X509Certificates.X509FindType.FindByThumbprint,
    "THUMBPRINT");
    try
    {
        //Izmantojam maksāšanas pieprasījuma identifikatoru no
        SavePaymentRequest()
        client.CreateInvoice(paymentRequestId, name, code, address);
    }
    catch (FaultException<PaymentRequestFault> ex)
    {
        error = ex.Detail.Message;
    }
```

Dokumenta kods: VDAA-13_7_17_41-VISS_2016-MM-II	Datums:08.10.2024	Versija:2.27
Datne: VISS_2016.II.MM_LVP3_v.2.27_tiraksts	Izstrādāja: M.Bambītis	Lpp.: 25 (35)
Valsts iepirkuma identifikācijas numurs VRAA/2010/10/ERAF/AK		

```

        client.Abort();
    }

```

PHP izmantošanas koda piemērs:

```

<?php
header('Content-Type: text/html; charset=utf-8');

$wsdlUrl = 'https://tv-
vraa.zzdats.lv/services/VRAA.VISS2.MM.Endpoint/v1.0/WCFX509/PaymentRequestService.svc?singleWsdl';
// Sertifikātu no pfx uz pem konvertē ar komandu:
// openssl pkcs12 -in fails.pfx -out fails.pem
$authCertPath = '/path/to/mm.pem';
$authCertPass = 'demo';

$client = new SoapClient($wsdlUrl, array('soap_version' => SOAP_1_1,
'local_cert' => $authCertPath, 'passphrase' => $authCertPass));

$paymentRequestId = '99e083d2-688f-481a-bb85-420eb11e6927';
$name = 'Ernestis Paraudziņš';
$code = '31129910101';
$address = 'Elizabetes 41/43, Rīga';

try
{
    $client-> CreateInvoice(array(
        'id' => $paymentRequestId,
        'name' => $name,
        'code' => $code,
        'address' => $address
    ));
}
catch(SoapFault $ex)
{
    die('SoapFault: ' . $ex->getMessage());
}

```

2.2.7. Maksāšanas pieprasījuma atcelšana

.NET izmantošanas koda piemērs:

```

PaymentRequestServiceReference.PaymentRequestServiceClient client =
    new
PaymentRequestServiceReference.PaymentRequestServiceClient();
string error = null;
Client.ClientCredentials.ClientCertificate.SetCertificate(
    System.Security.Cryptography.X509Certificates.StoreLocation.LocalMachine,
    System.Security.Cryptography.X509Certificates.StoreName.My,
    System.Security.Cryptography.X509Certificates.X509FindType.FindByThumbprint,
    "THUMBPRINT");
try
{
    //Izmantojam maksāšanas pieprasījuma identifikatoru no
SavePaymentRequest()
}

```

Dokumenta kods: VDAA-13_7_17_41-VISS_2016-MM-II	Datums:08.10.2024	Versija:2.27
Datne: VISS_2016.II.MM_LVP3_v.2.27_tiraksts	Izstrādāja: M.Bambītis	Lpp.: 26 (35)
Valsts iepirkuma identifikācijas numurs VRAA/2010/10/ERAF/AK		

```

        client.CancelPaymentRequest(paymentRequestId);
    }
    catch (FaultException<PaymentRequestFault> ex)
    {
        error = ex.Detail.Message;
        client.Abort();
    }
}

```

PHP izmantošanas koda piemērs:

```

<?php
header('Content-Type: text/html; charset=utf-8');

$wsdlUrl = 'https://tv-
vraa.zzdati.lv/services/VRAA.VISS2.MM.Endpoint/v1.0/WCFX509/PaymentRequestService.svc?singleWsdl';
// Sertifikātu no pfx uz pem konvertē ar komandu:
// openssl pkcs12 -in fails.pfx -out fails.pem
$authCertPath = '/path/to/mm.pem';
$authCertPass = 'demo';

$client = new SoapClient($wsdlUrl, array('soap_version' => SOAP_1_1,
'local_cert' => $authCertPath, 'passphrase' => $authCertPass));

$paymentRequestId = '99e083d2-688f-481a-bb85-420eb11e6927';

try
{
    $client-> CancelPaymentRequest(array(
        'paymentRequestId' => $paymentRequestId
    ));
}
catch(SoapFault $ex)
{
    die('SoapFault: ' . $ex->getMessage());
}

```

2.3. Maksāšanas procesa tīmekļa saskarne

Saskarne nodrošina maksāšanas līdzekļa izvēli, lietotāja pārsūtīšanu uz internetbanku un maksājumu statusu attēlošanu. Saskarnei var padot šādus papildus parametrus:

- fs: fonta izmērs, kādā attēlot MM interfeisu. Pieļaujamās vērtības: font-size-100, font-size-125, font-size-150
- language: valoda, kādā attēlot MM interfeisu. Pieļaujamās vērtības: LAT, ENG
- bm: vājredzīgo režīms, kurā attēlot MM interfeisu. Pieļaujamās vērtības: black-on-yellow, yellow-on-black

Piemērs: https://mmodulistest.vraa.gov.lv/MM_WebUI/WebUI2/Payments/?id=xxxxx&bm=yellow-on-black&fs=font-size-150&language=ENG

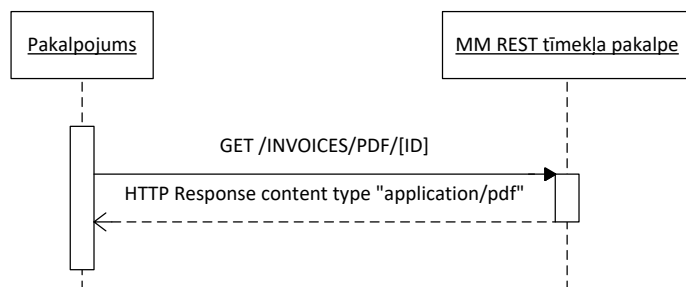
Pēc atgriešanās no internetbankas pēc maksājuma veikšanas, ja visi maksāšanas pieprasījumā esošie maksājumi ir apmaksāti, pie pogas “Atgriezties pie pakalpojuma sniedzēja” tiek attēlots skaitītājs, kas pēc konfigurējama sekunžu skaita automātiski novirza lietotāju atpakaļ uz epakalpojumu (maksāšanas pieprasījuma saglabāšanas parametrs ReturnRedirectUrl).

Dokumenta kods: VDAA-13_7_17_41-VISS_2016-MM-II	Datums: 08.10.2024	Versija: 2.27
Datne: VISS_2016.II.MM_LVP3_v.2.27_tiraksts	Izstrādāja: M.Bambītis	Lpp.: 27 (35)
Valsts iepirkuma identifikācijas numurs VRAA/2010/10/ERAF/AK		

2.4. Maksāšanas pieprasījuma attaisnojuma dokumenta datnes izgūšana PDF formātā – REST serviss

Pieslēdzoties tiek pārbaudīts klienta sertifikāts. Sertifikātu izsniedz pakalpojuma turētājs – VDAA, tam jābūt reģistrētam MM sertifikātu administrēšanas rīkā. Sertifikātam jābūt uzstādītam direktorijā Local Machine\Personal\Certificates. Jābūt uzstādītiem arī saknes sertifikātiem, lai klienta sertifikāts validētos. Saknes sertifikātiem jābūt uzstādītiem direktorijā Local Machine\Trusted Root Certification Authorities\Certificates.

Secību diagramma:



.NET izmantošanas koda piemērs:

```

namespace MM {
    using System.Net.Http;
    using System.Security.Cryptography.X509Certificates;

    ...
    Uri m_BaseAddress = new Uri("https://tv-
vraa.zzdati.lv/services/VRAA.VISS2.MM.Endpoint/v1.0/RESTX509/Invoices/");
    //maksāšanas pieprasījuma identifikators
    string prid = "08840043-1c60-e211-b74f-0050569b000e";

    WebRequestHandler handler = new WebRequestHandler();
    handler.ClientCertificateOptions = ClientCertificateOption.Manual;
    handler.ClientCertificates.Add(GetClientCertificate());

    HttpClient clientWithClientCert = new HttpClient(handler);
    clientWithClientCert.BaseAddress = m_BaseAddress;
    HttpResponseMessage response = clientWithClientCert.GetAsync("Pdf/" +
prid).Result;
    if (response.StatusCode == System.Net.HttpStatusCode.OK)
    {
        byte[] content = response.Content.ReadAsByteArrayAsync().Result;
        string contentType =
response.Content.Headers.ContentType.MediaType;
        //rezultāts
        ...
    }
    else
    {
        //kļūda
        ...
    }
    ...
}
  
```

Dokumenta kods: VDAA-13_7_17_41-VISS_2016-MM-II	Datums: 08.10.2024	Versija: 2.27
Datne: VISS_2016.II.MM_LVP3_v.2.27_tiraksts	Izstrādāja: M.Bambītis	Lpp.: 28 (35)
Valsts iepirkuma identifikācijas numurs VRAA/2010/10/ERAF/AK		

```
private static X509Certificate GetClientCertificate()
{
    X509Store store = new X509Store(StoreName.My,
StoreLocation.CurrentUser);
    store.Open(OpenFlags.ReadOnly);
    X509CertificateCollection col =
store.Certificates.Find(X509FindType.FindByThumbprint, "4a a8 de ce 0e 37 a2 61
8d ab a7 22 a7 b9 9e 9b e6 2c 9d 32", false);
    if (col.Count == 1)
    {
        return col[0];
    }
    else
    {
        //kļūda
        ...
    }
}
```

PHP izmantošanas koda piemērs:

```
<?php
$url = 'https://tv-
vraa.zzdati.lv/services/VRAA.VISS2.MM.Endpoint/v1.0/RESTX509/Invoices/Pdf/';

// Sertifikātu no pfx uz pem konvertē ar komandu:
// openssl pkcs12 -in fails.pfx -out fails.pem
$authCertPath = '/path/to/mm.pem';
$authCertPass = 'demo';

$paymentRequestId = '99e083d2-688f-481a-bb85-420eb11e6927';

$ch = curl_init($url . $paymentRequestId);

//curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_SSLCERT, $authCertPath);
curl_setopt($ch, CURLOPT_SSLCERTPASSWD, $authCertPass);

try
{
    $data = curl_exec($ch);
    if (curl_errno($ch) != CURLE_OK)
    {
        // TODO: Kļūdu apstrāde
        die(curl_error($ch));
    }
}
catch (Exception $ex)
{
    // TODO: Kļūdu apstrāde
    die($ex->getMessage());
}

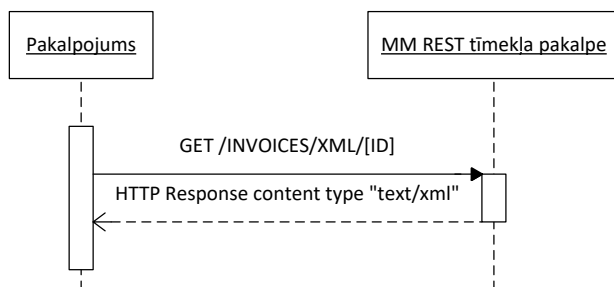
curl_close($ch);

header('Content-Type: application/pdf');
die($data);
```

2.5. Maksāšanas pieprasījuma attaisnojuma dokumenta datu izgūšana XML formātā – REST serviss

Pieslēdzoties tiek pārbaudīts klienta sertifikāts. Sertifikātu izsniedz pakalpojuma turētājs – VDAA, tam jābūt reģistrētam MM sertifikātu administrēšanas rīkā. Sertifikātam jābūt uzstādītam direktorijā Local Machine\Personal\Certificates. Jābūt uzstādītiem arī saknes sertifikātiem, lai klienta sertifikāts validētos. Saknes sertifikātiem jābūt uzstādītiem direktorijā Local Machine\Trusted Root Certification Authorities\Certificates.

Secību diagramma:



.NET izmantošanas koda piemērs:

```

namespace MM {
    using System.Net.Http;
    using System.Security.Cryptography.X509Certificates;

    ...
    Uri m_BaseAddress = new Uri("https://tv-
vraa.zzdati.lv/services/VRAA.VISS2.MM.Endpoint/v1.0/RESTX509/Invoices/");
    //maksāšanas pieprasījuma identifikators
    string prid = "08840043-1c60-e211-b74f-0050569b000e";

    WebRequestHandler handler = new WebRequestHandler();
    handler.ClientCertificateOptions = ClientCertificateOption.Manual;
    handler.ClientCertificates.Add(GetClientCertificate());

    HttpClient clientWithClientCert = new HttpClient(handler);
    clientWithClientCert.BaseAddress = m_BaseAddress;
    HttpResponseMessage response = clientWithClientCert.GetAsync("Xml/" +
prid).Result;
    if (response.StatusCode == System.Net.HttpStatusCode.OK)
    {
        byte[] content = response.Content.ReadAsByteArrayAsync().Result;
        string contentType =
response.Content.Headers.ContentType.MediaType;
        //rezultāts
        ...
    }
    else
    {
        //kļūda
        ...
    }
    ...
}
  
```

```
private static X509Certificate GetClientCertificate()
{
    X509Store store = new X509Store(StoreName.My,
StoreLocation.CurrentUser);
    store.Open(OpenFlags.ReadOnly);
    X509CertificateCollection col =
store.Certificates.Find(X509FindType.FindByThumbprint, "4a a8 de ce 0e 37 a2 61
8d ab a7 22 a7 b9 9e 9b e6 2c 9d 32", false);
    if (col.Count == 1)
    {
        return col[0];
    }
    else
    {
        //kļūda
        ...
    }
}
```

PHP izmantošanas koda piemērs:

```
<?php
$url = 'https://tv-
vraa.zzdats.lv/services/VRAA.VISS2.MM.Endpoint/v1.0/RESTX509/Invoices/Xml/';

// Sertifikātu no pfx uz pem konvertē ar komandu:
// openssl pkcs12 -in fails.pfx -out fails.pem
$authCertPath = '/path/to/mm.pem';
$authCertPass = 'demo';

$paymentRequestId = '99e083d2-688f-481a-bb85-420eb11e6927';

$ch = curl_init($url . $paymentRequestId);

//curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_SSLCERT, $authCertPath);
curl_setopt($ch, CURLOPT_SSLCERTPASSWD, $authCertPass);

try
{
    $data = curl_exec($ch);
    if (curl_errno($ch) != CURLE_OK)
    {
        // TODO: Kļūdu apstrāde
        die(curl_error($ch));
    }
}
catch (Exception $ex)
{
    // TODO: Kļūdu apstrāde
    die($ex->getMessage());
}

curl_close($ch);

header('Content-Type: text/xml; charset=utf-8');
die($data);
```

2.6. Maksāšanas pieprasījuma attaisnojuma dokumenta datnes izgūšana PDF formātā – SOAP serviss

Lai izgūtu maksāšanas pieprasījuma attaisnojuma dokumenta datnes nepieciešama loma MM_INVOICE_SERVICE.

Detalizētu informāciju par PFAS Auth aizsargātu pakalpju izsaukšanu skat. [2].

.NET izmantošanas koda piemērs:

```
/// <param name="paymentRequestId">MM maksāšanas pieprasījuma
identifikators</param>
/// <param name="xmlFormat">vai dokuments būs XML formātā (Default=PDF)</param>
/// <returns>DocumentDataContract</returns>
GetInvoice(Guid paymentRequestId, bool? xmlFormat = null)
```

2.7. Maksāšanas pieprasījuma attaisnojuma dokumenta datu izgūšana XML formātā – SOAP serviss

Lai izgūtu maksāšanas pieprasījuma attaisnojuma dokumenta datnes nepieciešama loma MM_INVOICE_SERVICE.

Detalizētu informāciju par PFAS Auth aizsargātu pakalpju izsaukšanu skat. [2].

.NET izmantošanas koda piemērs:

```
/// <param name="paymentRequestId">MM maksāšanas pieprasījuma
identifikators</param>
/// <param name="xmlFormat">vai dokuments būs XML formātā (Default=PDF)</param>
/// <returns>DocumentDataContract</returns>
GetInvoice(Guid paymentRequestId, bool? xmlFormat = null)
```

2.8. Maksāšanas pieprasījuma attaisnojuma dokumenta datu izgūšana PDF vai XML formātā – SOAP serviss ar sertifikāta autorizāciju bez lomu pārbaudes

Pieslēdzoties tiek pārbaudīts klienta sertifikāts. Sertifikātu izsniedz pakalpojuma turētājs – VDAA, tam jābūt reģistrētam MM sertifikātu administrēšanas rīkā. Sertifikātam jābūt uzstādītam direktoriājā Local Machine\Personal\Certificates. Jābūt uzstādītiem arī saknes sertifikātiem, lai klienta sertifikāts validētos. Saknes sertifikātiem jābūt uzstādītiem direktoriājā Local Machine\Trusted Root Certification Authorities\Certificates.

.NET izmantošanas koda piemērs:

```
SOAPServiceReference.MMServiceClient client =
    new SOAPServiceReference.MMServiceClient();

//Izmantojam maksāšanas pieprasījuma identifikatoru no SavePaymentRequest()
// otrais parametrs nosaka, vai atgriezt dokumentu XML formātā
string result = client.GetInvoice(paymentRequestId, false);
//result = objekts ar atribūtiem Data- (faila saturn baitu masīvā), Name- faila
nosaukums un ContentType- faila MIME tips)
```

Klienta konfigurācijas piemērs:

Dokumenta kods: VDAA-13_7_17_41 -VISS_2016-MM-II	Datums:08.10.2024	Versija:2.27
Datne: VISS_2016.II.MM_LVP3_v.2.27_tīrāksts	Izstrādāja: M.Bambītis	Lpp.: 32 (35)
Valsts iepirkuma identifikācijas numurs VRAA/2010/10/ERAF/AK		


```
<system.serviceModel>
  <bindings>
    <basicHttpBinding>
      <binding name="MMService" maxBufferPoolSize="2147483647"
        maxReceivedMessageSize="2147483647">
        <security mode="Transport">
          <transport clientCredentialType="Certificate" />
        </security>
      </binding>
    </basicHttpBinding>
  </bindings>

  <behaviors>
    <endpointBehaviors>
      <behavior name="EndpointBehavior">
        <dataContractSerializer maxItemsInObjectGraph="2147483647" />
      </behavior>
      <behavior name="MMCertificate">
        <clientCredentials>
          <clientCertificate findValue="sertifikāta-thumbprint" storeName="My"
            storeLocation="LocalMachine" x509FindType="FindByThumbprint" />
        </clientCredentials>
      </behavior>
    </endpointBehaviors>
  </behaviors>

  <client>
    <endpoint address="https://tv-
vraa.zzdats.lv/services/MM2/Invoice.SOAP/MMService.svc"
      behaviorConfiguration="MMCertificate" binding="basicHttpBinding"
        bindingConfiguration="MMService" contract="SOAPLVP3.MMService"
        name="MMService" />
    </client>
  </system.serviceModel>
```

PHP izmantošanas koda piemērs:

```
<?php
header('Content-Type: text/html; charset=utf-8');

$wsdlUrl = 'https://tv-
vraa.zzdats.lv/services/VRAA.VISS2.MM.Endpoint/v1.0/MMSOAP/MMService.svc?singleW
sdl';
// Sertifikātu no pfx uz pem konvertē ar komandu:
// openssl pkcs12 -in fails.pfx -out fails.pem
$authCertPath = '/path/to/mm.pem';
$authCertPass = 'demo';

$client = new SoapClient($wsdlUrl, array('soap_version' => SOAP_1_1,
'local_cert' => $authCertPath, 'passphrase' => $authCertPass));

try
{
    $resp = $client->GetInvoice(array(
```

```
'paymentRequestId' => '99e083d2-688f-481a-bb85-420eb11e6927',
'xmlFormat' => false
    ));

    echo "Content Type: " . $result->DocumentDataContract->ContentType .
"\n";
    echo "Name: " . $result->DocumentDataContract->Name . "\n";
    // Apstrādā bināros datus pēc vajadzības (atribūts $result-
>DocumentDataContract->Name)
}
catch(SoapFault $ex)
{
    // TODO: Nomainīt uz kļūdu apstrādi
    die('SoapFault: ' . $ex->getMessage());
}
```

3. Pielikumi

3.1. Maksāšanas pieprasījuma piemērs

```
<?xml version="1.0" encoding="utf-8"?>
<PaymentsDataStructure xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <PaymentRequest>
    <AllowedBanks>
      <AllowedBankStructure>
        <SwiftCode>DEMO0000</SwiftCode>
      </AllowedBankStructure>
      <AllowedBankStructure>
        <SwiftCode>HABALV22</SwiftCode>
      </AllowedBankStructure>
    </AllowedBanks>
    <ClientInfo>
      <ClientAddress>Rātslaukums 1, LV-1050, Rīga</ClientAddress>
      <ClientCode>31129900000</ClientCode>
      <ClientEMail>e@p.lv</ClientEMail>
      <ClientFirstName>Ernests</ClientFirstName>
      <ClientName>Paraudziņš</ClientName>
      <ClientPhone>26620000</ClientPhone>
    </ClientInfo>
    <ExpiryDate>2013-02-15</ExpiryDate>
    <ExpiryDateSpecified>true</ExpiryDateSpecified>
    <InvoiceIssueDate>2013-01-20</InvoiceIssueDate>
    <InvoiceIssueDateSpecified>true</InvoiceIssueDateSpecified>
    <NoInvoice>true</NoInvoice>
    <ParentPaymentRequestID i:nil="true"/>
    <PaymentObject>Test 12</PaymentObject>
    <PriceList>
      <PriceList>
        <Item>1</Item>
        <ItemElementName>Amount</ItemElementName>
        <Position>1</Position>
      </PriceList>
      <PriceList>
        <Item>2</Item>
        <ItemElementName>Quantity</ItemElementName>
        <Position>2</Position>
      </PriceList>
    </PriceList>
    <ResponseUrl>http://jusu-response-url</ResponseUrl>
    <ServerResponseUrl>http://jusu-response-url-s2s</ServerResponseUrl>
    <ServiceStep>00001</ServiceStep>
    <SwiftCode xsi:type="xsd:string">DEMO0000</SwiftCode>
  </PaymentRequest>
</PaymentsDataStructure>
```