Valsts reģionālās attīstības aģentūra

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

VRAA----

20.12.2022 versija 2.20

Rīgā 2022

 Dokumenta identifikācija

|  |  |
| --- | --- |
| Dokumenta ID: | VRAA-----V2.20-20.12.2022 |
| Dokumenta nosaukums: | Valsts informācijas sistēmu savietotājs (VISS) un Vienotā valsts un pašvaldību pakalpojumu portāla www.latvija.lv pilnveidošana un uzturēšana.Maksājumu modulis un Iedzīvotāju maksājumu nodrošināšanaIntegrācijas instrukcija. |
| Dokumenta kods: | VRAA---- |
| Versija: | Versija 2.20, Laidiens 20.12.2022 (saīsināti V2.20 20.12.2022) |

Saskaņojumi

|  |  |  |  |
| --- | --- | --- | --- |
| Organizācija | Vārds, uzvārds, amats  | Datums | Paraksts |
| Valsts reģionālās 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”SIA “ZZ Dats” | E.Vasiļevskis, Izstrādātāja pārstāvisA.Ušackis, Izstrādātāja pārstāvis | 02.02.202120.12.2022 |  |

Izmaiņu vēsture

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Versija | Datums | Apraksts | Organizācija | Autors |
| 2.02 | 07.03.2013. | Izstradāta 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, 2.3Dzēsta nodaļa 2.4 | SIA "ZZ Dats" | M.Gūtmanis |
| 2.06 | 11.12.2013. | Papildināta nodaļas 2.3, 2.4, 2.5, 2.6Izveidotas nodaļas 2.7, 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 2.4 un 2.9 | SIA “ZZ Dats” | M.Gūtmanis |
| 2.10 | 11.10.2016. | Papildināts .NET piemērs nodaļā 2.3 | SIA “ZZ Dats” | E.Vasiļevskis |
| 2.11 | 18.01.2017. | Precizētas nodaļas 2.3, 2.4, 2.5, 2.6 un 2.9 | 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 | 20.12.2022 | Precizētas nodaļas 2.2 | SIA “ZZ Dats” | K.Krūmiņliepa |

**Satura rādītājs**

[1. Ievads 6](#_Toc46916061)

[1.1. Dokumenta nolūks 6](#_Toc46916062)

[1.2. Darbības sfēra 6](#_Toc46916063)

[1.3. Termini un pieņemtie apzīmējumi 6](#_Toc46916064)

[1.4. Saistība ar citiem dokumentiem 6](#_Toc46916065)

[1.5. Dokumenta pārskats 7](#_Toc46916066)

[2. Izmantošanas scenāriji 8](#_Toc46916067)

[2.1. Portāla latvija.lv pakalpojumu asinhronas apmaksas veikšana 8](#_Toc46916068)

[2.2. Portāla latvija.lv pakalpojumu sinhronas apmaksas veikšana 8](#_Toc46916069)

[2.3. Ārēju pakalpojumu apmaksas veikšana 9](#_Toc46916070)

[2.4. Maksāšanas pieprasījuma statusa iegūšana ārējiem pakalpojumiem 12](#_Toc46916071)

[2.5. Maksāšanas pieprasījuma attaisnojuma dokumenta datnes izgūšana PDF formātā – REST serviss 14](#_Toc46916072)

[2.6. Maksāšanas pieprasījuma attaisnojuma dokumenta datu izgūšana XML formātā – REST serviss 16](#_Toc46916073)

[2.7. Maksāšanas pieprasījuma attaisnojuma dokumenta datnes izgūšana PDF formātā – SOAP serviss 18](#_Toc46916074)

[2.8. Maksāšanas pieprasījuma attaisnojuma dokumenta datu izgūšana XML formātā – SOAP serviss 18](#_Toc46916075)

[2.9. Maksāšanas pieprasījuma statusa iegūšana latvija.lv pakalpojumiem 18](#_Toc46916076)

[2.10. MM izsaukumi, izmantojot Lvp.EservicePlatform.Backend.PaymentApi mikroservisu 19](#_Toc46916077)

[2.10.1. Maksāšanas pieprasījuma izveidošana 19](#_Toc46916078)

[2.10.2. Maksāšanas pieprasījuma statusa pārbaude 21](#_Toc46916079)

[2.10.3. Maksāšanas pieprasījumu izgūšana 22](#_Toc46916080)

[2.10.4. Maksājumu izgūšana 23](#_Toc46916081)

[2.10.5. Maksājuma statusa pārbaude 25](#_Toc46916082)

[2.10.6. Maksāšanas pieprasījumu statusu pārbaude un apmaksas veida izgūšana 26](#_Toc46916083)

[2.10.7. Rēķina izveidošana 27](#_Toc46916084)

[3. Pielikumi 29](#_Toc46916085)

[3.1. Maksāšanas pieprasījuma piemērs 29](#_Toc46916086)

# Ievads

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

## Dokumenta nolūks

Dokuments sniedz detalizētu tehnisku informāciju par Maksājumu moduļa un Iedzīvotāju 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.

## Darbības sfēra

Tīmekļa pakalpju izsaukumi no ārējiem un PPK pakalpojumiem tiek veikti izmantojot X.509 sertifikātu. Latvija.lv pakalpojumi tīmekļa pakalpes izsaukumi tiek veikti izmantojot VISS integrācijas servisu.

## Termini un pieņemtie apzīmējumi

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

## 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. Meklētājs. Programmatūras prasību specifikācija (VRAA-6\_15\_11\_58-VISS\_2010-MKL-PPS);
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. Terminu un saīsinājumu indekss. (VRAA-6\_15\_11\_58-VISS\_2010-TSI);
3. „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).
4. 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 ar tehnisko specifikāciju". Maksājumu modulis un Iedzīvotāju maksājumu nodrošināšana Programmatūras prasību specifikācija (VRAA-13\_7\_17\_41-VISS\_2016-MM-PPS).
5. Valsts pārvaldes pakalpojuma kartītes aizpildes metodika (PK-MET-V.3.2-17.08.2018)

## Dokumenta pārskats

Dokuments sastāv no šādien nodalījumiem:

* 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.

# **Izmantošanas scenāriji**

Iespējamie MM izmantošanas scenāriji

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

## Portāla latvija.lv pakalpojumu asinhronas apmaksas veikšana

1. Pakalpojums izsauc integrācijas servisu apmaksas procesa uzsākšanai un nodod IVISRequest, kurš satur maksāšanas pieprasījumu. Tiek izveidots maksāšanas pieprasījums un klientam tiek nosūtīts paziņojums par nepieciešamību veikt apmaksu.
2. Klients KDV attaisnojumu dokumentu sarkstā atrod neapmaksātu maksāšanas pieprasījumu un veic apmaksu.
3. Pēc apmaksas veikšanas klients tiek atgriezts uz ResponseURL hipersaiti. Klients apmaksu var veikt līdz apmaksas termiņa iestāšanās laikam.
4. Tiek izveidota atbilde, kura satur maksāšanas pieprasījuma statusu, un nosūtīts IVISResponse.
5. Papildus maksāšanas pieprasījuma statusu un attaisnojuma dokumentu tīmekļa pakalpē var pieprasīt jebkurā brīdī pēc izveidošanas.

Secību diagramma:



## Portāla latvija.lv pakalpojumu sinhronas apmaksas veikšana

1. Pakalpojums izsauc integrācijas servisu apmaksas procesa uzsākšanai un nodod IVISRequest, 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 šādus URL 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

1. Pēc apmaksas veikšanas klients tiek atgriezts uz ResponseURL hipersaiti. Klients apmaksu var veikt līdz apmaksas termiņa iestāšanās laikam.
2. Pēc pādējā maksājuma statusa iegūšanas no apmaksas nodrošinātāja tiek izsaukta izveidots jauns pieprasījums, kurš satur maksāšanas pieprasījuma statusu, un nosūtīts IVISRequest.
3. Papildus maksāšanas pieprasījuma statusu un attaisnojuma dokumentu var pieprasīt jebkurā brīdī pēc izveidošanas.

Secību diagramma:



## Ārēju pakalpojumu apmaksas veikšana

1. Pieslēdzoties tiek pārbaudīts klienta sertifikāts. Sertifikātu izsniedz pakalpojuma turētājs – VRAA. 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. Tiek atgriezts maksāšanas pieprasījuma identifikators un lietotāja interfeisa izsaukšanas hipersaite.
3. Pakalpojumi izsauc atgrieztā MM lietotāja interfeisa hipersaiti. Hipersaitei atļauts pievienot 2.2 nodaļas 2. punktā minētos URL parametrus.
4. Pēc apmaksas veikšanas klients tiek atgriezts uz ResponseURL hipersaiti. Klients apmaksu var veikt līdz apmaksas termiņa iestāšanās laikam.
5. Pēc pādējā maksājuma statusa iegūšanas no apmaksas nodrošinātāja vai pēc apmaksas termiņa iestāšanās, neapmaksāta maksāšanas pieprasījuma gadījumā, tiek izsaukta ServerResponseURL hipesaite paziņojuma par maksāšanas pieprasījuma statusa maiņu nodošanai.
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.

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",

 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.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));

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',

 '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());

}

## Maksāšanas pieprasījuma statusa iegūšana ārējiem pakalpojumiem

1. Pieslēdzoties tiek pārbaudīts klienta sertifikāts. Sertifikātu izsniedz pakalpojuma turētājs – VRAA. 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:

PaymentRequestServiceReference.PaymentRequestServiceClient client =

 new PaymentRequestServiceReference.PaymentRequestServiceClient();

 string error = null;

 try

 {

 //Izmantojam maksāšanas pieprasījuma identifikatoru no SavePaymentRequest()

 string result = client.CheckPaymentStatus (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.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';

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());

}

## Maksāšanas pieprasījuma attaisnojuma dokumenta datnes izgūšana PDF formātā – REST serviss

1. Pieslēdzoties tiek pārbaudīts klienta sertifikāts. Sertifikātu izsniedz pakalpojuma turētājs – VRAA. 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.zzdats.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

 ...

 }

 ...

 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/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);

## Maksāšanas pieprasījuma attaisnojuma dokumenta datu izgūšana XML formātā – REST serviss

1. Pieslēdzoties tiek pārbaudīts klienta sertifikāts. Sertifikātu izsniedz pakalpojuma turētājs – VRAA. 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.zzdats.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);

## Maksāšanas pieprasījuma attaisnojuma dokumenta datnes izgūšana PDF formātā – SOAP serviss

1. Lai izgūtu maksāšanas pieprasījuma attaisnojuma dokumenta datnes nepieciešama loma MM\_INVOICE\_SERVICE.

Detalizētu informāciju par PFAS Auth aizsargātu pakalpju izsaukšanu skat. [3].

.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)

## Maksāšanas pieprasījuma attaisnojuma dokumenta datu izgūšana XML formātā – SOAP serviss

1. Lai izgūtu maksāšanas pieprasījuma attaisnojuma dokumenta datnes nepieciešama loma MM\_INVOICE\_SERVICE.

Detalizētu informāciju par PFAS Auth aizsargātu pakalpju izsaukšanu skat. [3].

.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)

## Maksāšanas pieprasījuma statusa iegūšana latvija.lv pakalpojumiem

1. Lai izgūtu maksāšanas pieprasījuma statusu nepieciešama loma MM\_PAYMENTREQUEST\_STATUS\_SERVICE.

Secību diagramma:



Detalizētu informāciju par PFAS Auth aizsargātu pakalpju izsaukšanu skat. [3].

.NET izmantošanas koda piemērs:

/// <param name=”paymentRequestId”>MM maksāšanas pieprasījuma identifikators</param>

/// <returns>Maksāšanas pieprasījuma statuss (P – Procesā, E = Izpildīts, R = ExpireDate, O = Other, iespējams tikai ja apmaksas nodrošinātājs atgriež)</returns>

GetPaymentRequestStatus(Guid paymentRequestId)

##  MM izsaukumi, izmantojot Lvp.EservicePlatform.Backend.PaymentApi mikroservisu

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

* **MM\_Cert** (klienta sertifikāta (skat.2.3. 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 MM PPS (VRAA-13\_7\_17\_41-VISS\_2016-MM-PPS [4], 4.1. nodaļai). Kļūdas teksts (ja tāds ir) tiek atgriezts MM-ErrorMessage header parametrā URL Encoded formātā.

###  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 (VRAA-13\_7\_17\_41-VISS\_2016-MM-PPS) [4]. Par cenrāža pozīciju un pakalpojuma soļu (PriceList un ServiceStep) vērtībām skatīt PPK aizpildes metodiku (PK-MET-V.3.2-17.08.2018)[5]

**Piemērs**

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.zzdats.lv/MMTests/S2S/?id=",

 "ServerResponseUrl": "https://tv-vraa.zzdats.lv/MMTests/S2S/ok1?id=",

 "ServiceStep" : "03G10",

 "SwiftCode" : "DEMO0000"

 }

 }

}

Izvaddati

PaymentRequestResponseStructure JSON objekts (atbilstoši [4] 3.2.2.1 nodaļai), ar laukiem:

* Maksāšanas pieprasījuma identifikators
* Maksājumu moduļa izsaukšanas URL. Hipersaitei atļauts pievienot 2.2 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](https://tv-vraa.zzdats.lv/sites/VRAA.VISS2.MM/WebUI.New/Payments/?id=05ae69d8-2be1-4a63-ae93-9d0e42e254f1)

}

###  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": "MIIM1QIBAzCCDJEGCSqGSIb3DQEHAaCCDIIEggx+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

###  Maksāšanas pieprasījumu izgūšana

|  |  |
| --- | --- |
| Identifikators | LvpContext.Payments.GetPaymentRequests |
| Apraksts | Atgriež maksāšanas pieprasījumu sarakstu |

metodes izsaukšana

**Adrese**

POST /api/v1/payments/getpaymentrequests

**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 – maksajuma summa līdz

OnlyNotPayed – tikai neapmaksātos pieprasījumus

Currency – maksājuma valūta

**Piemērs**

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

{

 "MM\_Cert": "MIIM1QIBAzCCDJEGCSqGSIb3DQEHAaCCDIIEggx+MII ... P63AgIH0A==",

 "CreatedDateFrom": "2020-01-31T12:41:00+3:00",

 "CreatedDateTill": "2020-08-14T12:41:00+3:00",

 "AmountFrom": "7.99",

 "OnlyNotPayed": false

}

Izvaddati

Maksāšanas pieprasījumu saraksts JSON formātā body daļā (atbilstoši [4] 3.2.2.9 nodaļai).

**Kļūdas**

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

**Piemērs**

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":"VRAA",

 "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": "VRAA",

 "ServiceName":"Pakalpojuma nosaukums 2",

 "TotalAmount":8.20,

 "Currency":"EUR",

 "UsageStatus": "Izmantots"

}

]

###  Maksājumu izgūšana

|  |  |
| --- | --- |
| Identifikators | LvpContext.Payments.GetPayments |
| Apraksts | Atgriež maksājumu sarkastu |

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 – maksajuma summa līdz

Currency – maksājuma valūta

**Piemērs**

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 [4] 3.2.2.10 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"

}

]

###  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**

* 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": "MIIM1QIBAzCCDJEGCSqGSIb3DQEHAaCCDIIEggx+MII ... P63AgIH0A==",

 "paymentRequestId": "a2a4b66a-a0b7-4b1d-bcc3-3d5a8d30b2b1",

 "position": "172"

}

Izvaddati

PaymentStatusResponseStructure JSON objekts (atbilstoši [4] 3.2.2.17 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"

}

###  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**

* 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": "MIIM1QIBAzCCDJEGCSqGSIb3DQEHAaCCDIIEggx+MII ... P63AgIH0A==",

 "paymentRequestId": "a2a4b66a-a0b7-4b1d-bcc3-3d5a8d30b2b1"

}

Izvaddati

PaymentRequestStatusResponseStructure JSON objekts (atbilstoši [4] 3.2.2.16 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"

}

###  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.

**QUERY parametri**

Nav.

**BODY parametri**

Id – maksāšanas pieprasījuma unikalais 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": "MIIM1QIBAzCCDJEGCSqGSIb3DQEHAaCCDIIEggx+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

}

# Pielikumi

## 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>