

Számlázz.hu

Invoice Agent

Technical specification

2018.07.18. – ver. 3.4

Content

Basics	4
What is Invoice Agent?	4
How Invoice Agent works?	4
Agent User	5
Error handling	5
Do we need to handle session cookies?	5
How can we test Invoice Agent?	5
What are the requirements of your system?	6
What kind of error messages can we get (error codes)?	6
How secure is this connection?	8
Can one XML file contains the details of more invoices?	8
What is necessary to create e-invoices?	8
XML validation using XSD	8
How can we provide more e-mail addresses?	9
When the email will be sending exactly?	9
How can I send the XML in the given requests?	9
Use for multiple websites	9
Generating invoice	10
Post request	10
Example XML file with explanation	10
VAT rates	12
XSD scheme compliance	12
What answers do we get during invoice creation?	12
Text response	12
XML response	13
Net, Gross and VAT values displayed on invoice	15
Rounding based on NET value (HUF invoice)	15
Rounding based on GROSS value (HUF invoice)	15
How can we know the number, GROSS and NET value of an invoice?	16
Can we create electronic invoices?	16
Agent output types	16
Will the system send an account notification after invoice generated by Invoice Agent?	17
Is there any example code for Invoice Agent calls?	17
Cancelling an invoice (Storno)	18
Data of post request	18

Example XML	18
XSD scheme compliance	18
How can we know the number, GROSS and NET value of a storno invoice?	19
What answers do we get during storno invoice creation?	19
Registering a credit entry	20
Data of post request	20
Example XML	21
XSD scheme compliance	21
What answers do we get during registering a credit entry?	22
Invoice payment feedback for the webshop	23
Querying the invoice PDF	24
Data of post request	24
Example XML	24
XSD scheme compliance	24
What answers do we get during PDF request?	24
PDF response	25
XML response	25
Querying the invoice XML	26
Deleting a pro forma invoice	26

1 Basics

First we would like to share some basic information about the Invoice Agent calls, after that we will detail each and every calls.

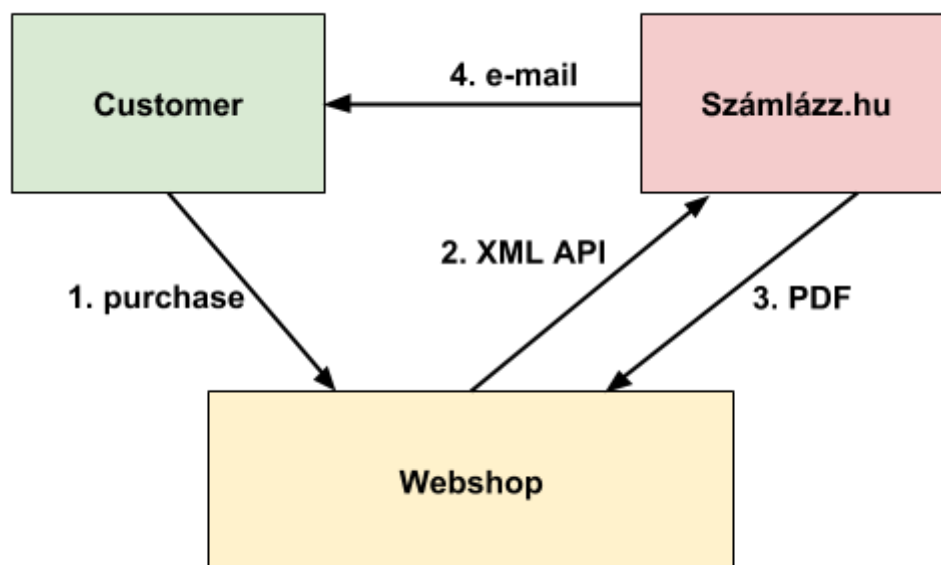
1.1 What is Invoice Agent?

The Invoice Agent is a non-browser-based interface of the Számlázz.hu system. It receives XML messages and depending on the received information it can:

- create invoices;
- storno invoices;
- switch the status of the invoices;
- create a PDF file of the invoice;

1.2 How Invoice Agent works?

The working mechanism of the system is displayed on the following picture



1. Buyer want to purchase something in the web shop;
2. The webshop sends an XML file in a post request to Számlázz.hu system.
3. Invoice Agent receives the XML file and generates the invoice and according to configuration it sends back the PDF invoice, a message like "xmlagentresponse=DONE;invoicenumber" or if there was any error, the description of the error;

4. If necessary, Számlázz.hu can send e-mails with PDF invoices as well;

1.3 Agent User

A Számlázz.hu company can be accessed by multiple users. It is advisable to create a separate user for the Agent with „accounting” privileges. The e-mail address of this user will receive Agent usage error messages.

1.4 Error handling

IMPORTANT! It has many reasons if the invoice agent does not create an invoice. Those reasons will not disappear without changing the calling parameters. Therefore if call is unsuccessful then recalling is prohibited. The maximum limit to send the same request to Invoice Agent is five times. If all of those requests are unsuccessful then please send an email to the operator of your webshop, and wait for his/her answer.

The Invoice Agent was not responsible for any of failed invoice in the past. If someone not complies with these rules, or causes big server load without any reasons, then the parties who call the Agent will banned.

1.5 Do we need to handle session cookies?

Yes, Invoice Agent is optimized in a way that the caller side must take care of storing the cookie and make sure that it is sent to Invoice Agent during the call. From the view of webshop must seem to be one “user”.

The process in details:

1. First call towards Invoice Agent, there is no session cookie on webshop, XML file sent to Invoice Agent, Invoice Agent perceives that there is no cookie yet so it generates a new session and a session cookie, reads the XML, etc.
2. First answer: one PDF file in response body, few other info in response header that contains the cookie as well on a name JSESSIONID;
3. Second call towards Invoice Agent: send the XML as for the first time and in the request header include the session cookie as well;
4. Second answer: one PDF in response body, and the same session cookie in the header, but with this webshop do not have to do anything;

After that only the 3rd and 4th steps get repeated. In case there is no Invoice Agent calls from a session cookie for 90 minutes, Számlázz.hu system deletes this cookie. For the next time there is a call towards Invoice Agent, a new cookie will be generated.

In light of these the following working mechanism is recommended in the codes calling Invoice Agent:

1. To call the Invoice Agent we need an XML file and a cookie file, if there is no cookie yet, call the Agent without it;
2. When the answer gets back, we need to save the cookie file and the PDF invoice;

1.6 How can we test Invoice Agent?

During testing we have to make sure not to create big amounts of invoices in testing environment (max 100 invoices per hour). In case we need some kind of stress test we can only do it by generating paper invoices, not PDF ones in order to avoid to overloading the timestamp server. For test accounts please contact the customer service at Számlázz.hu.

1.7 What are the requirements of your system?

Your system need to send an XML file in an HTTPS POST request to Számlázz.hu. This XML file must contains the controlling parameters and the data which necessary to perform the given Invoice Agent function. Do not forget to handle the session cookies.

The request/response pairs of given Invoice Agent call types (new invoice, reverse invoice, payments, ask PDF) will be described later.

1.8 What kind of error messages can we get (error codes)?

The blue text is the given error message (Hungarian). The translated error message is green. The remaining (black) text is the further explanation.

CODE	DESCRIPTION
1	<p>“Rendszerkarbantartás, kérem próbálja meg pár perc múlva.” System Maintenance, please try again in a few minutes.</p> <p>Internal error in Invoice Agent. Webshop does not have to do anything, maintenance team of Számlázz.hu is notified and take action.</p>
3	<p>“Sikertelen bejelentkezés” Login error</p> <p>Provided login name and password not accepted.</p>
49	<p>“Elektronikus számla készítéséhez kérem adja meg a kulcstartót nyitó titkos jelszavát.” To create an electronic invoice, please give your secret password which opens the keystore.</p> <p>To generate the invoice please provide the secret password for key.</p>
53	<p>“Hiányzó XML fájl.” Missing XML file.</p> <p>Probably the XML was not sent as file, but as an input content of an HTML.</p>
54	<p>“E-számla készítés nincs engedélyezve” E-invoice generation is not permitted.</p> <p>Probably your subscription package does not contain it, or there is no own certificate uploaded, in such case you have to accept that KBOSS.hu Kft. Can use its own certificate for the invoices.</p>
55	<p>“E-számla aláírása sikertelen.” E-invoice signature was unsuccessful.</p> <p>Either your certificate expired or timestamp server could not be reached.</p>

56	<p>“Számleaértesítő küldése sikertelen.”</p> <p>Invoice was saved, but no e-mail was sent.</p> <p>In such case the Webshop will get a notification e-mail so we can send it manually.</p>
57	<p>“XML beolvasási hiba.”</p> <p>XML reading error.</p> <p>There is an error in the sent XML file. The response body contains more information.</p>
135	<p>“Számlea Agent futtatásához kérjük böngészőkében lépjen ki a Számlázz.hu rendszerből”</p> <p>In order to run Invoice Agent, please log out from Számlázz.hu in your browser.</p> <p>This can happen, in case we are logged in to Számlázz.hu in one browser window and in another we are testing the XML with test HTML.</p>
136	<p>“ Bejelentkezési hiba ... Kérjük lépjen be a Számlázz.hu rendszerbe böngészőn keresztül. “</p> <p>Login error ... Please log in to Számlázz.hu's system via browser.</p> <p>For some reason we cannot use Invoice Agent. Reasons might be the following:</p> <ul style="list-style-type: none"> - subscription expired - there is a pending invoice towards Számlázz.hu - there is a delay in payment <p>In these cases please log in to the Számlázz.hu and check your subscription in the menu “Díjcsomagom”.</p>
164	<p>“Ezt a funkciót csak olyan felhasználó veheti igénybe, aki egyetlen Számlázz.hu-s fiókhoz férhet hozzá.
Több fiókhoz történő hozzáférés kizárja a funkció használatát.”</p> <p>“The user is allowed to access only one company’s billing account. If the user got access more than one billing account, the operation will fail.”</p> <p>The username given in the xml file should have access only one company’s billing account.</p>
202	<p>“A megadott számlaszám előtag nem megfelelő. Agent használat során csak a Számlázz.hu rendszerében korábban regisztrált előtagokat lehet használni.”</p> <p>“The given invoice prefix is not usable. You can only use those prefixes for the Invoice Agent that were previously registered in Szamlazz.hu’s system.”</p> <p>You need to login to the website and add the desired prefixes before you could use them with the Invoice Agent.</p>
259	<p>“ A tétel nettó értéke nem megfelelő; nettó érték = nettó egységár x mennyiség. Termék: “</p> <p>The NET value of the item is not correct; NET value = NET unit price * quantity. Product:</p> <p>Your values don’t fit into this formula, please verify and correct the NET value. The response body contains the name of the wrong product after the “Termék” word.</p>

260	<p>“ A tétel áfa értéke nem megfelelő; áfa érték = tétel nettó értéke x áfakulcs mértéke / 100. Termék: “</p> <p>The VAT value of the item is not correct; VAT value = NET price of item * VAT rate / 100. Product:</p> <p>Your values don't fit into this formula, please verify and correct the VAT value. The response body contains the name of the wrong product after the “Termék” word.</p>
261	<p>“ A tétel bruttó értéke nem megfelelő; bruttó érték = tétel nettó értéke + tétel áfa értéke. Termék: “</p> <p>The gross value of the item is not correct; gross value = NET value of item + VAT value of item. Product:</p> <p>Your values don't fit into this formula, please verify and correct the gross value. The response body contains the name of the wrong product after the “Termék” word.</p>
262	<p>“ A tétel nettó értéke nem megfelelő; nettó érték = nettó egységár x mennyiség. Termék: “</p> <p>The NET value of the item is not correct; NET value = NET unit price * quantity. Product:</p> <p>Your values don't fit into this formula, please verify and correct the NET value. The response body contains the row number of the wrong product after the “Termék” word.</p>
263	<p>“ A tétel áfa értéke nem megfelelő; áfa érték = tétel nettó értéke x áfakulcs mértéke / 100. Termék: “</p> <p>The VAT value of the item is not correct; VAT value = NET price of item * VAT rate / 100. Product:</p> <p>Your values don't fit into this formula, please verify and correct the VAT value. The response body contains the row number of the wrong product after the “Termék” word.</p>
264	<p>“ A tétel bruttó értéke nem megfelelő; bruttó érték = tétel nettó értéke + tétel áfa értéke. Termék: “</p> <p>The gross value of the item is not correct; gross value = NET value of item + VAT value of item. Product:</p> <p>Your values don't fit into this formula, please verify and correct the gross value. The response body contains the row number of the wrong product after the “Termék” word.</p>

1.9 How secure is this connection?

The connection is built up on a 256 bit encrypted SSL connection.

1.10 Can one XML file contains the details of more invoices?

No. Each XML file must contain the information of one invoice.

1.11 What is necessary to create e-invoices?

In order to create e-invoices we need our own certificate, but we can also use the certificate of Számlázz.hu.

IMPORTANT, to use the Számlázz.hu certificate leave the password for key empty in the XML file, or simply omit the <kulcstartojelszo> tag.

1.12 XML validation using XSD

We can validate the content of the XML file with an XSD file. In the XSD file there is a description that describes what must be in the given file.

For validation we use the xmllint software that can be installed as a package on Linux and can be downloaded for Windows on the following link:

<http://code.google.com/xmllint/>

1.13 How can we provide more e-mail addresses?

In case we would like to provide more e-mail addresses in the XML file's e-mail field, they must be separated by "," (comma).

1.14 When the email will be sending exactly?

In the case of test account the system send an invoice note, but not to the email address which is given in the XML, rather to the email address which was set in the settings (the "Email cím" value in this form:

<https://www.szamlazz.hu/szamla/?action=directlogin&targetpage=beallitasokstep1> .

If you set an email address in the buyer's data, then the system will send the invoice to this address. Obviously if you are not set any email address, the system won't send the invoice. This is the default behavior.

You can define the buyer's email address in the XML, and ask for Invoice Agent to NOT send the invoice to the buyer. In this case you must add this line just after the email address: <sendEmail>>false</sendEmail>

1.15 How can I send the XML in the given requests?

You need to send the requests to the same URL every time. This is the following:

<https://www.szamlazz.hu/szamla/>. The invoice Agent decides which function to perform via the name of the field which contains the XML file in the HTTPS request. The following table contains the names which we can use to perform a given function.

1.16 Use for multiple websites

Multiple websites can use the same Számlázz.hu user for the Agent. If you'd like to separate invoices by webshop please configure a separate invoice „prefix” for each of them.

2 Generating invoice

2.1 Post request

The post request looks like to following code:

```
<html>
<head><meta content="text/html; CHARSET=UTF-8"></head>
<body>
  <form action="https://www.szamlazz.hu/szamlaz" method="post" enctype="multipart/form-data">
    <input type="file" name="action-xmlagentxmlfile"><br>
    <input type="submit" name="generate" value="Számla generálás">
  </form>
</body>
</html>
```

The request must be "multipart/form-data" type and must contain one file on the name "action-xmlagentxmlfile".

2.2 Example XML file with explanation

In the below example all fields are mandatory, but some of them can be left blank, as they are not used in Invoice Agent only in other interfaces of Számlázz.hu. Számlázz.hu does not make any calculations on the given amount so it is mandatory to provide all amount that must be displayed on the invoice.

Invoice agent will run some checking:

quantity * net unit price == net price

the calculated VAT amount

net price + VAT == gross price

If it finds any error, it will send a notification (error code: 57, 259, 260, 261, 262, 263, 264) and it won't generate the invoice.

In the XML the order of the fields is strict, they cannot be switched.

```
<?xml version="1.0" encoding="UTF-8"?>
<xmlszamla xmlns="http://www.szamlazz.hu/xmlszamla"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.szamlazz.hu/xmlszamla xmlszamla.xsd ">

<beallitasok> <!-- settings -->
  <felhasznalo>teszt01</felhasznalo> <!-- a Számlázz.hu user -->
  <jelszo>teszt01</jelszo> <!-- a Számlázz.hu's user password -->
  <eszamla>true</eszamla> <!-- „true” in case we need to create an e-invoice -->
  <kulcstartojelszo>teszt01</kulcstartojelszo> <!-- in case of e-invoice this is the password
for the keyholder. Otherwise omit tis tag -->
  <szamlaLetoltes>true</szamlaLetoltes> <!-- „true” in case we would like to get the
PDF invoice int he response -->
  <valaszVerzio>1</valaszVerzio> <!-- 1: gives a simple text or PDF as answer.
2: xml answer, in case we asked for the PDF as well,
it will be included int he XML with base64 coding.
-->
  <aggregator></aggregator> <!-- omit this tag -->
</beallitasok>

<fejlec> <!-- header -->
```

```

<keltDatum>2010-09-12</keltDatum> <!-- creating date, in this exact format -->
<teljesitesDatum>2010-09-10</teljesitesDatum> <!-- payment date -->
<fizetesiHataridoDatum>2010-09-20</fizetesiHataridoDatum> <!-- due date -->
<fizmod>Átutalás</fizmod> <!-- payment type: it can be seen in case we create the invoice
                                from browser -->
<penznem>HUF</penznem> <!-- currency: it can be seen in case we create the invoice
                                from browser -->
<szamlaNyelve>hu</szamlaNyelve> <!-- language of invoice, can be: de, en, it, hu, fr, ro, sk, hr
                                -->
<megjegyzes>Számla megjegyzés</megjegyzes> <!-- comment of the invoice -->
<arfolyamBank>MNB</arfolyamBank> <!-- name of bank: in case of invoice about other currency
                                than HUF we have to display which bank's exchange rates
                                did we use to calculate VAT -->
<arfolyam>0.0</arfolyam> <!-- exchange rate: in case of invoice about other currency
                                than HUF we have to display which bank's exchange rates
                                did we use to calculate VAT -->
<rendelesSzam></rendelesSzam> <!-- order number -->
<elolegszamla>false</elolegszamla> <!-- deposit invoice -->
<vegszamla>false</vegszamla> <!-- invoice (after a deposit invoice) -->
<helyesbitoszamla>false</helyesbitoszamla> <!-- correction invoice -->
<helyesbitettSzamlaszam></helyesbitettSzamlaszam> <!-- the number of the corrected invoice -->
<dijbekero>false</dijbekero> <!-- proform invoice -->
<szamlaszamElotag></szamlaszamElotag> <!-- One of the prefixes from the invoice pad menu -->
</fejlec>

<elado> <!-- Details of the merchant-->
  <bank>BB</bank> <!-- bank name -->
  <bankszamlaszam>11111111-22222222-33333333</bankszamlaszam> <!-- bank account -->
  <emailReplyto></emailReplyto> <!-- reply e-mail address -->
  <emailTargy>Számla értesítő</emailTargy> <!-- e-mail subject -->
  <emailSzoveg>mail text</emailSzoveg> <!-- text of e-mail -->
</elado>

<vevo> <!--Buyer details -->
  <nev>Kovacs Bt.</nev> <!-- name -->
  <irsz>2030</irsz> <!-- ZIP code -->
  <telepules>Érd</telepules> <!-- city -->
  <cim>Tárnoki út 23.</cim> <!-- address -->
  <email>buyer@example.com</email> <!-- e-mail address, if given, we will send the invoice
                                to this mail address -->
  <sendEmail>false</sendEmail> <!-- should we send the email to the customer (by email) -->
  <adoszam>12345678-1-42</adoszam> <!-- fiscal number/tax number -->
  <postazasiNev>Kovács Bt. postázási név</postazasiNev> <!--delivery name/postal name -->
  <postazasiIrsz>2040</postazasiIrsz> <!--delivery ZIP code/postal ZIP code -->
  <postazasiTelepules>Budaörs</postazasiTelepules> <!--delivery city/postal city -->
  <postazasiCim>Szivárvány utca 8.</postazasiCim> <!--delivery address/postal address -->
  <telefonszam>Tel:+3630-555-55-55, Fax:+3623-555-555</telefonszam> <!-- phone number -->
  <megjegyzes>A portáról felszólni a 214-es mellékre.</megjegyzes> <!-- comment -->
</vevo>

<fuvarlevel> <!-- waybill, we do not need this: omit the entire tag -->
  <uticel></uticel>
  <futarSzolgalat></futarSzolgalat>
</fuvarlevel>

<tetelek> <!-- items on invoice -->
  <tetel> <!-- item 1 -->
    <megnevezes>Elado izé</megnevezes> <!-- name -->
    <mennyiseg>1.0</mennyiseg> <!-- quantity -->
    <mennyisegiEgyseg>db</mennyisegiEgyseg> <!-- unit of quantity -->
    <nettoEgysegar>10000</nettoEgysegar> <!-- unit price -->
    <afakulcs>27</afakulcs> <!-- VAT rate -->
    <nettoErtek>10000.0</nettoErtek> <!-- net price -->
    <afaErtek>2700.0</afaErtek> <!-- VAT amount -->
    <bruttoErtek>12700.0</bruttoErtek> <!-- gross amount -->
    <megjegyzes>lorem ipsum</megjegyzes> <!-- comment for item -->
  </tetel>

```

```

<tetel> <!-- item 2, details are same as above -->
  <megnevezes>Elado izé 2</megnevezes>
  <mennyiseg>2.0</mennyiseg>
  <mennyisegiEgyseg>db</mennyisegiEgyseg>
  <nettoEgysegar>10000</nettoEgysegar>
  <afakulcs>27</afakulcs>
  <nettoErtek>20000.0</nettoErtek>
  <afaErtek>5400.0</afaErtek>
  <bruttoErtek>25400.0</bruttoErtek>
  <megjegyzes>lorem ipsum 2</megjegyzes>
</tetel>
</tetelek>

</xmlszamla>

```

2.3 VAT rates

In the XML file into the <afakulcs></afakulcs> field we have to write one of the following values:

Value in XML field	Description
TAM	supply exempt from VAT / exempt supply
AAM	person exempt from VAT / exempt person
EU	Within EU
EUK	Outside EU
MAA	exempt from tax
F.AFA	reverse VAT
K.AFA	differential VAT
0, 5, 7, 18, 19, 20, 25, 27	exact VAT rate

2.4 XSD scheme compliance

The sent XML file must comply with the following XSD scheme. The fields where we see minOccurs="0" are not mandatory.

You can find the XSD file in this zip file: <https://www.szamlazz.hu/szamla/docs/SzamlaAgent.zip>

2.5 What answers do we get during invoice creation?

The received response depends on the value of the `valaszVerzio` variable sent in the request.

- in case it is 1 or not provided, than the response might be:
 - text – if <valaszVerzio>1</valaszVerzio>
 - PDF file – if <valaszVerzio>2</valaszVerzio>
- in case it is 2 than the answer will be XML in all cases;

2.5.1 Text response

There might be three different kind of answers. In the XML we can choose to receive the invoice in PDF format (<szamlaLetoltes> true/false </szamlaLetoltes>):

- if it is TRUE (so we need the option to download the PDF invoice) and everything went well, than the response will be the PDF that is the invoice itself;
- if it is FALSE (so we do not need the option to download the PDF invoice) and everything went well, than the answer is a text with the following structure:
 - o text indicating the success: "xmlagentresponse=DONE"
 - o semicolon ";"
 - o the number of created invoice, such as: "2011-123"

So the value of the response after a successful invoice creating should look like the following:

"xmlagentresponse=DONE;2011-123"

- in case there was an error during invoice creation than we will get one or more error messages

The format of the error message will be as can be seen below. The part between the [ERR] and ----- is the important one.

```
[ERR] Számla mentés sikertelen. Már létező rendelésszám: XX. ----- t.getMessage(): Már létező
rendelésszám: XX. ----- [CEG:978] [MODUL:SZAMLAZZGUI]
[INSTANCE:hu.kboss.szamlazz.gui.action.AcXmlSzamla] Már létező rendelésszám:
XX.[LOCATION:hu.kboss.szamlazz.api.bean.SzFej.insert (SzFej.java) #1700]
hu.kboss.szamlazz.api.bean.SzFej.insert (SzFej.java:1700)
hu.kboss.szamlazz.gui.action.AcXmlSzamla.performAction (AcXmlSzamla.java:242)
hu.kboss.szamlazz.controller.Controller.executeAction (Controller.java:715)
hu.kboss.szamlazz.controller.Controller.doGet (Controller.java:468)
hu.kboss.szamlazz.controller.Controller.doPost (Controller.java:639)
javax.servlet.http.HttpServlet.service (HttpServlet.java:641)
javax.servlet.http.HttpServlet.service (HttpServlet.java:722)
org.apache.catalina.core.ApplicationFilterChain.internalDoFilter (ApplicationFilterChain.java:305)
org.apache.catalina.core.ApplicationFilterChain.doFilter (ApplicationFilterChain.java:210)
org.apache.catalina.core.StandardWrapperValve.invoke (StandardWrapperValve.java:224)
org.apache.catalina.core.StandardContextValve.invoke (StandardContextValve.java:169)
org.apache.catalina.authenticator.AuthenticatorBase.invoke (AuthenticatorBase.java:472)
org.apache.catalina.core.StandardHostValve.invoke (StandardHostValve.java:168)
org.apache.catalina.valves.ErrorReportValve.invoke (ErrorReportValve.java:98)
org.apache.catalina.valves.AccessLogValve.invoke (AccessLogValve.java:927)
org.apache.catalina.core.StandardEngineValve.invoke (StandardEngineValve.java:118)
org.apache.catalina.connector.CoyoteAdapter.service (CoyoteAdapter.java:407)
org.apache.coyote.ajp.AjpProcessor.process (AjpProcessor.java:200)
org.apache.coyote.AbstractProtocol$AbstractConnectionHandler.process (AbstractProtocol.java:579)
org.apache.tomcat.util.net.JIoEndpoint$SocketProcessor.run (JIoEndpoint.java:307)
java.util.concurrent.ThreadPoolExecutor$Worker.runTask (ThreadPoolExecutor.java:886)
java.util.concurrent.ThreadPoolExecutor$Worker.run (ThreadPoolExecutor.java:908)
java.lang.Thread.run (Thread.java:662)
```

TIP: in case we request a paper-based non e-invoice creation and we need the download option (<szamlaletoltes>true</szamlaletoltes>) than we can indicate that we not only need the original version of the invoice but the copy as well in one PDF. This we can mark in the <szamlaletoltesPld>2</szamlaletoltesPld> variable.

2.5.2 XML response

The structure of the response is very different, but the logic behind it is similar.

In the header of the XML response we still get the information describe above.

In case the `<szamlaletoltes>` value is set to TRUE than the invoice PDF will appear within the `<PDF></PDF>` tags with base64 encryption.

XML response in case of success when `<szamlaletoltes>` is FALSE:

```
<?xml version="1.0" encoding="UTF-8"?>
<xmlszamlavalasz xmlns="http://www.szamlazz.hu/xmlszamlavalasz"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <sikeres>true</sikeres>
  <szamlaszam>XXX-2012-3</szamlaszam>
  <szamlanetto>30000</szamlanetto>
  <szamlabrutto>38100</szamlabrutto>
</xmlszamlavalasz>
```

XML response in case of success when `<szamlaletoltes>` is TRUE:

```
<?xml version="1.0" encoding="UTF-8"?>
<xmlszamlavalasz xmlns="http://www.szamlazz.hu/xmlszamlavalasz"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <sikeres>true</sikeres>
  <szamlaszam>XXX-2012-3</szamlaszam>
  <szamlanetto>30000</szamlanetto>
  <szamlabrutto>38100</szamlabrutto>
  <pdf>JVBERi0xLjQKJdDUxdGKMSAwIG9iago8PCAvUyAvR29UbyAvRCBbMiAwIFIGIC9GaXQgXSA+Pgpl
bmRvYmoKNCAwIG9iaia8PAovTGVuZ3RoIDE4MDggICAgICAKL0ZpbHRlciAvRmxhdGVEZWVvZGUK
Pj4Kc3RyZWFTcnJaxVjLbttGFN37K7gpQAH1ZN4P72zUTpvATVu72aRZqBLtKHqLEm0jAor8es88
RjESLduRgXoRTTjDO+fee+6LNLvJaPb66PTq6NU5c5kjTnOdXV1nmhJnMqMEMUJkV8PsQ85I7+PV
mlfntnGMCSKEg4xw4kuP2/wvykVPGp6Xk2FYK0rz/sxvfe0VQgicULRclM0yGmHJegVXxtH8crU5
Pp30l83/HvcKxrH66fXrglPGC+GxHNGEf/sXyuiMS2KsYQGmsUQLmxVcE6VtRHu56iW0SqlwZafM
....
MDAwMDg2NTIxIDAwdAwIG4gCjAwMDAwODY2NTcgMDAwMDAgbiAKMDAwMDA4NjY5MyAwMDAwMCBu
IAowMDAwMDg2Nzk4IDAwdAwIG4gCnRyYWlsZXIKPDwgLlNpemUgNDIKLlJvb3QgNDAgMCBSCi9J
bmZvIDQxIDAuUgovSUQgWzxxENDHERjNFQtdFM0MwQ0QyNDY0RUIzNzYwRTJBQzdBRj4gPEQ0OERG
MOVBN0UzQzBDRDI0NjRFQjM3NjBFMkFDN0FGP10gPj4Kc3RhcncR4cmVmCjg3MTIzCiUlRU9G</pdf>
</xmlszamlavalasz>
```

XML response in case of unsuccessful creation:

```
<?xml version="1.0" encoding="UTF-8"?>
<xmlszamlavalasz xmlns="http://www.szamlazz.hu/xmlszamlavalasz"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <sikeres>false</sikeres>
  <hibakod>3</hibakod>
  <hibauzenet><![CDATA[Bejelentkezési hiba - a megadott login név és jelszó pároshoz nem létezik felhasználó]]></hibauzenet>
</xmlszamlavalasz>
```

The response complies with the following XSD scheme:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.szamlazz.hu/xmlszamlavalasz"
xmlns:tns="http://www.szamlazz.hu/xmlszamlavalasz" elementFormDefault="qualified">
  <complexType name="szamlavalasz">
    <sequence>
      <element name="sikeres" type="boolean" maxOccurs="1" minOccurs="1"/>
      <element name="hibakod" type="string" maxOccurs="1" minOccurs="0"/>
      <element name="hibauzenet" type="string" maxOccurs="1" minOccurs="0"/>
      <element name="szamlaszam" type="string" maxOccurs="1" minOccurs="0"/>
      <element name="szamlanetto" type="double" maxOccurs="1" minOccurs="0"/>
    </sequence>
  </complexType>
</schema>
```

```

    <element name="szamlabrutto" type="double" maxOccurs="1" minOccurs="0"></element>
    <element name="pdf" type="base64Binary" maxOccurs="1" minOccurs="0"></element>
  </sequence>
</complexType>
<element name="xmlszamlavalasz" type="tns:szamlavalasz"></element>
</schema>

```

2.6 Net, Gross and VAT values displayed on invoice

Notions:

- Net unit price: the net price of one piece (or any other unit) of an item on the invoice;
- Quantity of item: the sold amount of the given item;
- Total value of item: the net unit price of the item multiplied by the quantity of the item;
- VAT value of item: calculated from Total value of item with the given VAT rate;
- Gross value of item: sum of Total value of item and VAT value of item;

In case of an invoice in other currency than HUF it is possible to use not whole amounts (eg: 12.45).

In case of HUF invoices the amounts of "VAT value of item" and "Total value of item" must be integers. IT means that in a lot of cases we have to round the amounts as we will barely get an integer using the 26% VAT rate.

There are two ways for rounding:

2.6.1 Rounding based on NET value (HUF invoice)

In case we are issuing invoice for a business partner and not a natural person (B2B), than we have to use this method in the following way:

- Take the "Net unit price" (it can be any amount integer or not).
- Multiply the "Net unit price" with the "Quantity of item". If the result is not an integer, we must round it. This will be the "Total net value of item".
- Take the "Total net value of item" and the VAT rate (eg: 27%) and calculate the VAT amount. If the result is not an integer, we must round it.
- Add the VAT value (integer) and the "Total net value of item" (integer) to get the Gross amount. Since both value were integers the result will be integer as well.

2.6.2 Rounding based on GROSS value (HUF invoice)

In case we are issuing an invoice to a natural person (B2C) we have to use this method.

- Take the "Gross unit price" (it can be any amount integer or not).
- Multiply the "Gross unit price" with the "Quantity of item". If the result is not an integer, we must round it. This will be the "Total gross value of item".

- From the "Total gross value of item" and the VAT rate calculate the VAT amount. The formula for the counting is: $\text{"Total gross value of item"} / (100 + \text{VAT rate}) \times \text{VAT RATE}$. Example: I sold 3 pieces of a book that has a 500 HUF GROSS price:
 - o Gross unit price: 500 HUF
 - o Quantity: 3 pieces
 - o Gross value of item: $500 \times 3 = 1500$ HUF
 - o VAT rate: 27%
 - o VAT value: $1500 / (100 + 27) \times 27 = 318,89$ HUF
rounding this amount will mean that the VAT amount of the invoice is 319 HUF
- The "Total net value of item" can be calculated by extracting the VAT amount from the "Total gross value of item". Since both values are integers, the result will be integer as well.

In case we do not keep these rules Invoice Agent will do the roundings in order to comply with the law.

2.7 How can we know the number, GROSS and NET value of an invoice?

All HTTP response consist of two parts. First is the header the second one is the body. In case we asked for the Invoice the body will be the PDF. The header is always a name=value pair. In all programming languages that support HTTP requests there is an option to get the values in the header. In Invoice Agent we can put data in the header on the following names, that always start with "szlahu_":

- szlahu_szamlaszam: the number of the created invoice, URL encoded
- szlahu_nettovegosszeg: number type data, not URL encoded
- szlahu_bruttovegosszeg: number type data, not URL encoded
- szlahu_error: text of error message, I sentence, URL encoded
- szlahu_error_code: number type data, not URL encoded

If there are error codes in the header, than the other date (invoice number, gross and net value) will not be present in the header. In case of XML response besides the header the XML will include these information as well.

2.8 Can we create electronic invoices?

Yes. Invoice Agent can generate both e-invoice and regular, paper based invoice. What kind of invoice will be created depends on the value of the "eszamla" variable. IN case of TRUE e-invoice will be created, in case of FALSE regular invoice will be created.

2.9 Agent output types

The Agent can make documents of the following types:

- invoice
- storno invoice (reverse invoice)
- deposit invoice based on payment request
- final invoice
- corrective invoice
- proforma (prepayment, deposit) request
- delivery note

2.10 Will the system send an account notification after invoice generated by Invoice Agent?

Only in case when the in the XML the e-mail address of the buyer is provided. Here we can also set the subject and content text of the notification e-mail. Also there are a few options to format the text with the help of some BBCode tag.

Accepted BBcodes are: [b], [/b], [i], [/i], [u], [/u], [h1], [/h1]...[h6],[/h6], [center], [/center]

2.11 Is there any example code for Invoice Agent calls?

Yes, PHP and Joomla Virtuemart example code is attached to the following ZIP file, called szamla_agent_minta.php (or you can download from here:

<https://www.szamlazz.hu/szamla/docs/SzamlaAgent.zip>).

.Net example is available at <https://szamlazdotnet.codeplex.com/> and Woocommerce at <https://wordpress.org/plugins/integration-for-szamlazzhu-woocommerce/>

3 Reversing an invoice (storno)

3.1 Post request data

Here is a short example HTML code for the invoice storno call for Invoice Agent:

```
<html>
  <head><meta content="text/html; CHARSET=UTF-8"></head>
  <body>
    <form action="https://www.szamlazz.hu/szamlaz" method="post" enctype="multipart/form-data">
      <input type="file" name="action-szamlaz_agent_st"><br>
      <input type="submit" name="generate" value="Számla sztornóztatása">
    </form>
  </body>
</html>
```

So the request must be "multipart/form-data" type and must contain one file on the name "action-szamlaz_agent_st".

3.2 Example XML

We have to use the following XML structure:

```
<?xml version="1.0" encoding="UTF-8"?>
<xmlszamlaz xmlns="http://www.szamlazz.hu/xmlszamlaz" xmlns:xsi="http://www.w3.org/2001/XMLSchemainstance"
xsi:schemaLocation="http://www.szamlazz.hu/xmlszamlaz xmlszamlaz.xsd ">
<beallitasok><!-- settings -->
  <felhasznalo>Kérem töltse ki!</felhasznalo><!-- a Számlázz.hu user -->
  <jelszo>Kérem töltse ki!</jelszo><!-- a Számlázz.hu password -->
  <eszamlaz>false</eszamlaz><!-- e-invoice (true/false) -->
  <szamlazLetoltes>true</szamlazLetoltes><!-- download PDF -->
  <szamlazLetoltesPld>1</szamlazLetoltesPld><!-- download PDF copy or original -->
</beallitasok>
<fejlec><!-- header -->
  <szamlaszam>E-TST-2011-1</szamlaszam><!-- invoice number -->
  <keltDatum>2010-09-12</keltDatum><!-- date of issue -->
  <teljesitesDatum>2010-09-10</teljesitesDatum><!-- date of fulfillment -->
  <tipus>SS</tipus><!-- type -->
</fejlec>
<elado><!-- seller -->
  <emailReplyto>elado@example.com</emailReplyto>
  <emailTargy>Email tárgya</emailTargy><!-- e-mail subject -->
  <emailSzoveg>Lorem ipsum</emailSzoveg><!-- e-mail content -->
</elado>
<vevo><!-- buyer -->
  <email>buyer@example.com</email>
</vevo>
</xmlszamlaz>
```

3.3 XSD scheme compliance

The sent XML file must comply to the following XSD scheme. The fields where we see that minOccurs="0" are not mandatory.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.szamlazz.hu/xmlszamlaz"
xmlns:tns="http://www.szamlazz.hu/xmlszamlaz" elementFormDefault="qualified">
<complexType name="vevoTipus">
  <sequence>
    <element name="email" type="string" maxOccurs="1" minOccurs="0"></element>
  </sequence>
</complexType>
<complexType name="eladoTipus">
  <sequence>
```

```

    <element name="emailReplyto" type="string" maxOccurs="1" minOccurs="0"></element>
    <element name="emailTargy" type="string" maxOccurs="1" minOccurs="0"></element>
    <element name="emailSzoveg" type="string" maxOccurs="1" minOccurs="0"></element>
  </sequence>
</complexType>
<complexType name="beallitasokTipus">
  <sequence>
    <element name="felhasznalo" type="string" maxOccurs="1" minOccurs="1"></element>
    <element name="jelszo" type="string" maxOccurs="1" minOccurs="1"></element>
    <element name="eszamla" type="boolean" maxOccurs="1" minOccurs="1"></element>
    <element name="kulcstartojelszo" type="string" maxOccurs="1" minOccurs="0"></element>
    <element name="szamlaLetoltes" type="boolean" maxOccurs="1" minOccurs="1"></element>
    <element name="szamlaLetoltesPld" type="int" maxOccurs="1" minOccurs="0"></element>
  </sequence>
</complexType>
<complexType name="fejlecTipus">
  <sequence>
    <element name="szamlaszam" type="string" maxOccurs="1" minOccurs="1"></element>
    <element name="keltDatum" type="date" maxOccurs="1" minOccurs="0"></element>
    <element name="teljesitesDatum" type="date" maxOccurs="1" minOccurs="0"></element>
    <element name="tipus" type="string" maxOccurs="1" minOccurs="0"></element>
  </sequence>
</complexType>
<element name="xmlszamlast">
  <complexType>
    <sequence>
      <element name="beallitasok" type="tns:beallitasokTipus" maxOccurs="1" minOccurs="1"></element>
      <element name="fejlec" type="tns:fejlecTipus" maxOccurs="1" minOccurs="1"></element>
      <element name="elado" type="tns:eladoTipus" maxOccurs="1" minOccurs="0"></element>
      <element name="vevo" type="tns:vevoTipus" maxOccurs="1" minOccurs="0"></element>
    </sequence>
  </complexType>
</element>
</schema>

```

3.4 How can we know the number, GROSS and NET value of a storno invoice?

All HTTP response consists of two parts. First is the header the second one is the body. In case we asked for the Invoice the body will be the PDF. The header is always a name=value pair. In all programming languages that support HTTP requests there is an option to get the values in the header. In Invoice Agent we can put data in the header on the following names, that always start with "szlahu_":

- szlahu_szamlaszam: the number of the created invoice, URL encoded
- szlahu_nettovegosszeg: number type data, not URL encoded
- szlahu_bruttovegosszeg: number type data, not URL encoded
- szlahu_error: text of error message, I sentence, URL encoded
- szlahu_error_code: number type data, not URL encoded

3.5 What answers do we get during storno invoice creation?

There might be three different kind of answers. In the XML we can choose to receive the invoice in PDF format (<szamlaLetoltes>true/false</szamlaLetoltes>):

- if it is TRUE (so we need the option to download the PDF invoice) and everything went well, than the response will be the PDF that is the invoice itself;

- if it is FALSE (so we do not need the option to download the PDF invoice) and everything went well, then the answer is a text with the following structure:
 - o text indicating the success: "xmlagentresponse=DONE"
 - o semicolon ";"
 - o the number of created invoice, such as: "2011-123"

So the value of the response after a successful invoice creating should look like the following:

"xmlagentresponse=DONE;2011-123"

- in case there was an error during invoice creation than we will get one or more error messages

The format of the error message will be as can be seen below. The part between the [ERR] and ----- is the important:

```
[ERR] Számla mentés sikertelen. Már létező rendelésszám: XX. ----- t.getMessage(): Már létező
rendelésszám:XX. ----- [CEG:978] [MODUL:SZAMLAZZGUI]
[INSTANCE:hu.kboss.szamlazz.gui.action.AcXmlSzamla] Már
létező rendelésszám: XX.[LOCATION:hu.kboss.szamlazz.api.bean.SzFej.insert (SzFej.java) #1700]
hu.kboss.szamlazz.api.bean.SzFej.insert (SzFej.java:1700)
hu.kboss.szamlazz.gui.action.AcXmlSzamla.performAction (AcXmlSzamla.java:242)
hu.kboss.szamlazz.controller.Controller.executeAction (Controller.java:715)
hu.kboss.szamlazz.controller.Controller.doGet (Controller.java:468)
hu.kboss.szamlazz.controller.Controller.doPost (Controller.java:639)
javax.servlet.http.HttpServlet.service (HttpServlet.java:641)
javax.servlet.http.HttpServlet.service (HttpServlet.java:722)
org.apache.catalina.core.ApplicationFilterChain.internalDoFilter (ApplicationFilterChain.java:305)
org.apache.catalina.core.ApplicationFilterChain.doFilter (ApplicationFilterChain.java:210)
org.apache.catalina.core.StandardWrapperValve.invoke (StandardWrapperValve.java:224)
org.apache.catalina.core.StandardContextValve.invoke (StandardContextValve.java:169)
org.apache.catalina.authenticator.AuthenticatorBase.invoke (AuthenticatorBase.java:472)
org.apache.catalina.core.StandardHostValve.invoke (StandardHostValve.java:168)
org.apache.catalina.valves.ErrorReportValve.invoke (ErrorReportValve.java:98)
org.apache.catalina.valves.AccessLogValve.invoke (AccessLogValve.java:927)
org.apache.catalina.core.StandardEngineValve.invoke (StandardEngineValve.java:118)
org.apache.catalina.connector.CoyoteAdapter.service (CoyoteAdapter.java:407)
org.apache.coyote.ajp.AjpProcessor.process (AjpProcessor.java:200)
org.apache.coyote.AbstractProtocol$AbstractConnectionHandler.process (AbstractProtocol.java:579)
org.apache.tomcat.util.net.JIoEndpoint$SocketProcessor.run (JIoEndpoint.java:307)
java.util.concurrent.ThreadPoolExecutor$Worker.runTask (ThreadPoolExecutor.java:886)
java.util.concurrent.ThreadPoolExecutor$Worker.run (ThreadPoolExecutor.java:908)
java.lang.Thread.run (Thread.java:662)
```

TIP: in case we request a paper-based non e-invoice creation and we need the download option (<szamlaletoltes>true</szamlaletoltes>) than we can indicate that we not only need the original version of the invoice but the copy as well in one PDF. This we can mark in the <szamlaletoltesPld>2</szamlaletoltesPld> variable.

4 Registering a credit entry

4.1 Data of post request

Here is a short example HTML code for the invoice storno call for Invoice Agent:

```
<html>
  <head><meta content="text/html; CHARSET=UTF-8"></head>
  <body>
    <form action="https://www.szamlazz.hu/szamlaz/" method="post" enctype="multipart/form-data">
      <input type="file" name="action-szamlaz_agent_kifiz"><br>
      <input type="submit" name="generate" value="Számla generálás">
    </form>
  </body>
</html>
```

So the request must be "multipart/form-data" type and must contain one file on the name "action-szamlaz_agent_kifiz".

4.2 Example XML

We have to use the following simple XML in the request in which we can register maximum 5 credit entry.

```
<?xml version="1.0" encoding="UTF-8"?>
<xmlszamlakifiz xmlns="http://www.szamlazz.hu/xmlszamlakifiz" xmlns:xsi="http://www.w3.org/2001/XMLSchemainstance"
xsi:schemaLocation="http://www.szamlazz.hu/xmlszamlakifiz xmlszamlakifiz.xsd">
  <beallitasok> <!-- settings -->
    <felhasznalo>Kérem töltse ki!</felhasznalo><!-- a Számlázz.hu user -->
    <jelszo>Kérem töltse ki!</jelszo><!-- a Számlázz.hu's user password -->
    <szamlaszam>E-TST-2011-1</szamlaszam> <!-- invoice number -->
    <additiv>>false</additiv> <!-- if it is true, then the former credit entries will be retained -->
  </beallitasok>
  <kifizetes> <!-- payment -->
    <datum>2012-01-01</datum><!-- date -->
    <jogcim>készpénz</jogcim><!-- payment type -->
    <osszeg>1000</osszeg><!-- amount -->
  </kifizetes>
  <kifizetes><!-- an other payment -->
    <datum>2012-01-15</datum>
    <jogcim>átutalás</jogcim>
    <osszeg>2000</osszeg>
    <leiras>Test description</leiras> <!-- Description, not required -->
  </kifizetes>
</xmlszamlakifiz>
```

4.3 XSD scheme compliance

The sent XML file must comply to the following XSD scheme. The fields where we see that minOccurs="0" are not mandatory.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.szamlazz.hu/xmlszamlakifiz"
xmlns:tns="http://www.szamlazz.hu/xmlszamlakifiz" elementFormDefault="qualified">
  <complexType name="beallitasokTipus">
    <sequence>
      <element name="felhasznalo" type="string" maxOccurs="1" minOccurs="1"></element>
      <element name="jelszo" type="string" maxOccurs="1" minOccurs="1"></element>
      <element name="szamlaszam" type="string" maxOccurs="1" minOccurs="1"></element>
      <element name="additiv" type="boolean" maxOccurs="1" minOccurs="1"></element>
    </sequence>
  </complexType>
  <complexType name="kifizetesTipus">
    <sequence>
      <element name="datum" type="date" maxOccurs="1" minOccurs="1"></element>
      <element name="jogcim" type="string" maxOccurs="1" minOccurs="1"></element>
      <element name="osszeg" type="double" maxOccurs="1" minOccurs="1"></element>
      <element name="leiras" type="string" maxOccurs="1" minOccurs="0"></element>
    </sequence>
  </complexType>
  <complexType name="szamlakifizTipus">
    <sequence>
```

```

<element name="beallitasok" type="tns:beallitasokTipus" maxOccurs="1" minOccurs="1"></element>
<element name="kifizetes" type="tns:kifizetesTipus" maxOccurs="5" minOccurs="0"></element>
</sequence>
</complexType>
<element name="xmlszamlakifiz" type="tns:szamlaKifizTipus"></element>
</schema>

```

4.4 What answers do we get during registering a credit entry?

All HTTP responses consist of two parts. First is the header the second one is the body. In case we asked for the Invoice the body will be the PDF. The header is always a name=value pair. In all programming languages that support HTTP requests there is an option to get the values in the header. In Invoice Agent we can put data in the header on the following names, that always start with "szlahu_":

- szlahu_szamlaszam: the number of the created invoice on which we register the credit entry, URL encoded
- szlahu_nettovegosszeg: number type data, not URL encoded
- szlahu_bruttovegosszeg: number type data, not URL encoded
- szlahu_error: text of error message, I sentence, URL encoded
- szlahu_error_code: number type data, not URL encoding

If there are error codes in the header, than the other data (invoice number, gross and net value) will not be present in the header. In case of XML response besides the header the XML will include these information as well.

There are two possible answers:

- in case everything went well, the response will be as follows:
 - text marking the success: "xmlagentresponse=DONE"
- in case there was some kind of error we will get an error message as follows:

The format of the error message will be as can be seen below. The part between the [ERR] and ----- is the important one.

```

[ERR] Számla mentés sikertelen. Már létező rendelésszám: XX. ----- t.getMessage(): Már létező
rendelésszám:
XX. ----- [CEG:978] [MODUL:SZAMLAZZGUI] [INSTANCE:hu.kboss.szamlazz.gui.action.AcXmlSzamla] Már
létező rendelésszám: XX. [LOCATION:hu.kboss.szamlazz.api.bean.SzFej.insert (SzFej.java) #1700]
hu.kboss.szamlazz.api.bean.SzFej.insert (SzFej.java:1700)
hu.kboss.szamlazz.gui.action.AcXmlSzamla.performAction (AcXmlSzamla.java:242)
hu.kboss.szamlazz.controller.Controller.executeAction (Controller.java:715)
hu.kboss.szamlazz.controller.Controller.doGet (Controller.java:468)
hu.kboss.szamlazz.controller.Controller.doPost (Controller.java:639)
javax.servlet.http.HttpServlet.service (HttpServlet.java:641)
javax.servlet.http.HttpServlet.service (HttpServlet.java:722)
org.apache.catalina.core.ApplicationFilterChain.internalDoFilter (ApplicationFilterChain.java:305)
org.apache.catalina.core.ApplicationFilterChain.doFilter (ApplicationFilterChain.java:210)
org.apache.catalina.core.StandardWrapperValve.invoke (StandardWrapperValve.java:224)
org.apache.catalina.core.StandardContextValve.invoke (StandardContextValve.java:169)
org.apache.catalina.authenticator.AuthenticatorBase.invoke (AuthenticatorBase.java:472)
org.apache.catalina.core.StandardHostValve.invoke (StandardHostValve.java:168)
org.apache.catalina.valves.ErrorReportValve.invoke (ErrorReportValve.java:98)
org.apache.catalina.valves.AccessLogValve.invoke (AccessLogValve.java:927)
org.apache.catalina.core.StandardEngineValve.invoke (StandardEngineValve.java:118)
org.apache.catalina.connector.CoyoteAdapter.service (CoyoteAdapter.java:407)
org.apache.coyote.ajp.AjpProcessor.process (AjpProcessor.java:200)

```

```
org.apache.coyote.AbstractProtocol$AbstractConnectionHandler.process(AbstractProtocol.java:579)
org.apache.tomcat.util.net.JIoEndpoint$SocketProcessor.run(JIoEndpoint.java:307)
java.util.concurrent.ThreadPoolExecutor$Worker.runTask(ThreadPoolExecutor.java:886)
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:908)
java.lang.Thread.run(Thread.java:662)
```

4.5 Invoice payment feedback for the webshop

If you need feedback, you can set it to the following link:

<https://www.szamlazz.hu/szamla/?action=directlogin&targetpage=beallitasokstep6>

In that case, if the state of payment changes then the Számlázz.hu send the following data:

- invoice number
- order number (if applicable)
- gross amount
- the paid amount.

Example:

http://www.webaruhaz.hu/kifizetetszamla?szlahu_szamlaszam=E-2015-123&szlahu_rendelesszam=RND1234&szlahu_bruttovegosszeg=10000&szlahu_kifizetettbrutto=10000

5 Querying the invoice PDF

The requested PDF can be 2 different kind:

- in case of regular, paper-based invoice it contains the next copy, or in case it has not been printed before it can be 1 or 2 copies depending on the settings;
- in case of e-invoice there are no such thing as copies, so it will give the original version back;

5.1 Data of post request

Here is a short example HTML code for the invoice storno call for Invoice Agent:

```
<html>
<head><meta content="text/html; CHARSET=UTF-8"></head>
<body>
  <form action="https://www.szamlazz.hu/szamlaz/" method="post" enctype="multipart/form-data">
    <input type="file" name="action-szamlaz_agent_pdf"><br>
    <input type="submit" name="generate" value="Számle pdf lekérés">
  </form>
</body>
</html>
```

So the request must be "multipart/form-data" type and must contain one file on the name "action-szamlaz_agent_pdf".

5.2 Example XML

```
<xmlszamlapdf xmlns="http://www.szamlazz.hu/xmlszamlapdf" xmlns:xsi="http://www.w3.org/2001/XMLSchemainstance"
xsi:schemaLocation="http://www.szamlazz.hu/xmlszamlapdf xmlszamlapdf.xsd">
  <felhasznalo>Kérem töltse ki!</felhasznalo><!-- a Számlázz.hu user -->
  <jelszo>Kérem töltse ki!</jelszo><!-- a Számlázz.hu password -->
  <szamlaszam>E-TST-2011-1</szamlaszam><!-- invoice number -->
  <valaszVerzio>2</valaszVerzio><!-- response type (see details below) -->
</xmlszamlapdf>
```

5.3 XSD scheme compliance

The sent XML file must comply to the following XSD scheme. The fields where we see that minOccurs="0" are not mandatory.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.szamlazz.hu/xmlszamlapdf"
xmlns:tns="http://www.szamlazz.hu/xmlszamlapdf" elementFormDefault="qualified">
  <complexType name="beallitasokTipus">
    <sequence>
      <element name="felhasznalo" type="string" maxOccurs="1" minOccurs="1"></element>
      <element name="jelszo" type="string" maxOccurs="1" minOccurs="1"></element>
      <element name="szamlaszam" type="string" maxOccurs="1" minOccurs="1"></element>
      <element name="valaszVerzio" type="int" maxOccurs="1" minOccurs="1"></element>
    </sequence>
  </complexType>
  <element name="xmlszamlapdf" type="tns:beallitasokTipus"></element>
</schema>
```

5.4 What answers do we get during PDF request?

The type of response depends on the value of the <valaszVerzio> variable sent in the request.

- if it is 1 or not set, the response will be a PDF file
- if it is 2, the response will be an XML file

In both cases the parameters will be in the header.

5.4.1 PDF response

There are two kind of answers:

- in case everything went well, the response is a PDF file, that is the invoice itself;
- in case there was an error we will get one or more error messages.

The format of the error message will be as can be seen below. The part between the [ERR] and ----- is the important one.

```
[ERR] Számla mentés sikertelen. Már létező rendelésszám: XX. ----- t.getMessage(): Már létező
rendelésszám:
XX. ----- [CEG:978] [MODUL:SZAMLAZZGUI] [INSTANCE:hu.kboss.szamlazz.gui.action.AcXmlSzamla] Már létező
rendelésszám: XX.[LOCATION:hu.kboss.szamlazz.api.bean.SzFej.insert (SzFej.java) #1700]
hu.kboss.szamlazz.api.bean.SzFej.insert (SzFej.java:1700)
hu.kboss.szamlazz.gui.action.AcXmlSzamla.performAction (AcXmlSzamla.java:242)
hu.kboss.szamlazz.controller.Controller.executeAction (Controller.java:715)
hu.kboss.szamlazz.controller.Controller.doGet (Controller.java:468)
hu.kboss.szamlazz.controller.Controller.doPost (Controller.java:639)
javax.servlet.http.HttpServlet.service (HttpServlet.java:641)
javax.servlet.http.HttpServlet.service (HttpServlet.java:722)
org.apache.catalina.core.ApplicationFilterChain.internalDoFilter (ApplicationFilterChain.java:305)
org.apache.catalina.core.ApplicationFilterChain.doFilter (ApplicationFilterChain.java:210)
org.apache.catalina.core.StandardWrapperValve.invoke (StandardWrapperValve.java:224)
org.apache.catalina.core.StandardContextValve.invoke (StandardContextValve.java:169)
org.apache.catalina.authenticator.AuthenticatorBase.invoke (AuthenticatorBase.java:472)
org.apache.catalina.core.StandardHostValve.invoke (StandardHostValve.java:168)
org.apache.catalina.valves.ErrorReportValve.invoke (ErrorReportValve.java:98)
org.apache.catalina.valves.AccessLogValve.invoke (AccessLogValve.java:927)
org.apache.catalina.core.StandardEngineValve.invoke (StandardEngineValve.java:118)
org.apache.catalina.connector.CoyoteAdapter.service (CoyoteAdapter.java:407)
org.apache.coyote.ajp.AjpProcessor.process (AjpProcessor.java:200)
org.apache.coyote.AbstractProtocol$AbstractConnectionHandler.process (AbstractProtocol.java:579)
org.apache.tomcat.util.net.JIoEndpoint$SocketProcessor.run (JIoEndpoint.java:307)
java.util.concurrent.ThreadPoolExecutor$Worker.runTask (ThreadPoolExecutor.java:886)
java.util.concurrent.ThreadPoolExecutor$Worker.run (ThreadPoolExecutor.java:908)
java.lang.Thread.run (Thread.java:662)
```

5.4.2 XML response

The structure of the response will be way different since it is an XML, but the logic behind it is similar.

Besides the XML in the header the above mentioned parameters are included as well. The invoice PDF is in the XML between the <pdf></pdf> tags with base64 encoding.

An example XML response in case of successful request:

```
<?xml version="1.0" encoding="UTF-8"?>
<xmlszamlavalasz xmlns="http://www.szamlazz.hu/xmlszamlavalasz"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <sikeres>true</sikeres>
  <szamlaszam>XXX-2012-3</szamlaszam>
  <szamlanetto>30000</szamlanetto>
  <szamlabrutto>38100</szamlabrutto>
  <pdf>JVBERi0xLjQKJdDUxdgKMSAwIG9iago8PCAvUyAvR29UbyAvRCBbMiAwIFRlIG9GaXQgXSA+Pgpl
bmRvYmoKNCAwIG9iaaiA8PAovTGVuZ3RoIDE4MDggICAgICAKL0ZpbHRlciAvRmxdGVEZWNvZGUK
Pj4Kc3RyZWVtCnjaxVjLbttGFN37K7gpQAHlZN4P72zUTpvATVu72aRZqBLtKHqLEm0jAor8es88
```

```
RJESLduRgXoRTTjDO+fee+6LNLvJaPb66PTq6NU5c5kjTnOdXV1nmhJnMqMEMUJkV8PsQ85I7+PV
m1fntnGMSKEg4xw4kuP2/wvykVPGp6Xk2FYK0rz/sxvfe0VQgicULRclMOyGmHJegVXxtH8crU5
Pp30l83/HvcKxrH66fXrglPGC+GxHNGEf/sXyuiMS2KsYQGmsUQLmxVcE6VtRHu56iW0SqlwZafM
....
MDAwMDg2NTIxIDAuMDAwIG4gCjAwMDAwODY2NTcgMDAwMDAgbiAKMDAwMDA4NjY5MyAwMDAwMCBu
IAowMDAwMDg2Nzk4IDAuMDAwIG4gCnRyYWlsZXIKPDwgL1NpemUgNDIKL1Jvb3QgNDAgMCBScI9J
bmZvIDQxIDAuUgovSUQgWzxEbDhERjNFQtdFM0MwQ0QyNDY0RUIzNzYwRtJBQzdBRj4gPEQ0OERG
M0VBN0UzQzBDRDI0NjRFQjM3NjBFMkFDN0FGPl0gPj4Kc3RhcnR4cmVmCjg3MTIzCiUlRU9G</pdf>
</xmlszamlavalasz>
```

An example response in case of unsuccessful request:

```
<?xml version="1.0" encoding="UTF-8"?>
<xmlszamlavalasz xmlns="http://www.szamlazz.hu/xmlszamlavalasz"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <sikeres>false</sikeres>
  <hibakod>3</hibakod>
  <hibauzenet><![CDATA[Bejelentkezési hiba - a megadott login név és jelszó párosához nem létezik
felhasználó]]></hibauzenet>
</xmlszamlavalasz>
```

The response complies with the following XSD:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.szamlazz.hu/xmlszamlavalasz"
xmlns:tns="http://www.szamlazz.hu/xmlszamlavalasz" elementFormDefault="qualified">
  <complexType name="szamlavalasz">
    <sequence>
      <element name="sikeres" type="int" maxOccurs="1" minOccurs="1"></element>
      <element name="hibakod" type="string" maxOccurs="1" minOccurs="0"></element>
      <element name="hibauzenet" type="string" maxOccurs="1" minOccurs="0"></element>
      <element name="szamlaszam" type="string" maxOccurs="1" minOccurs="0"></element>
      <element name="szamlanetto" type="double" maxOccurs="1" minOccurs="0"></element>
      <element name="szamlabrutto" type="double" maxOccurs="1" minOccurs="0"></element>
      <element name="pdf" type="base64Binary" maxOccurs="1" minOccurs="0"></element>
    </sequence>
  </complexType>
  <element name="xmlszamlavalasz" type="tns:szamlavalasz"></element>
</schema>
```

6 Querying the invoice XML

6.1 Post request data

```
<html>
<head><meta content="text/html; CHARSET=UTF-8"></head>
<body>
  <form action="https://www.szamlazz.hu/szamlaz" method="post" enctype="multipart/form-data">
    <input type="file" name="action-szamlaz_agent_xml"><br>
    <input type="submit" name="generate" value="Számla xml lekérés">
  </form>
</body>
</html>
```

6.2 Example XML

```
<?xml version="1.0" encoding="UTF-8"?>
<xmlsamlaxml xmlns="http://www.szamlazz.hu/xmlsamlaxml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.szamlazz.hu/xmlsamlaxml http://www.szamlazz.hu/docs/xsds/agentpdf/xmlsamlaxml.xsd">
  <felhasznalo>Kérem töltse ki!</felhasznalo>
  <jelszo>Kérem töltse ki!</jelszo>
  <szamlaszam>E-TST-2011-1</szamlaszam>
  <rendelesSzam>123</rendelesSzam>
  <pdf>true</pdf>
</xmlsamlaxml>
```

6.3 XSD scheme compliance

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.szamlazz.hu/xmlsamlaxml"
xmlns:tns="http://www.szamlazz.hu/xmlsamlaxml" elementFormDefault="qualified">
  <element name="xmlsamlaxml">
    <complexType>
      <sequence>
        <element name="felhasznalo" type="string" maxOccurs="1" minOccurs="1"></element>
        <element name="jelszo" type="string" maxOccurs="1" minOccurs="1"></element>
        <element name="szamlaszam" type="string" maxOccurs="1" minOccurs="0"></element>
        <element name="rendelesSzam" type="string" maxOccurs="1" minOccurs="0"></element>
        <element name="pdf" type="boolean" maxOccurs="1" minOccurs="0"></element>
      </sequence>
    </complexType>
  </element>
</schema>
```

6.4 Expected invoice data results

6.4.1 Example XML response in case of an unsuccessful request

```
<?xml version="1.0" encoding="UTF-8"?>
<szamla xmlns="http://www.szamlazz.hu/szamla" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.szamlazz.hu/szamla szamla.xsd">
  <szallito>
    <id>201</id>
    <nev>DEVE-1966 Lorem ipsum dolor sit amet, consectetur cs amet., 512356234</nev>
    <cim>
      <orszag>Magyarorsz&#225;g</orszag>
      <irsz>1086</irsz>
      <telepules>Budapest</telepules>
      <cim>Szerdahelyi utca 4-8</cim>
    </cim>
    <adoszam>202 336 0856</adoszam>
    <adoszameu>I&#268;DPH:SK2023360856, I&#268;O:46393218</adoszameu>
    <bank>
      <nev>UniCredit Bank Hungary Zrt.</nev>
      <bankszamla></bankszamla>
    </bank>
  </szallito>
  <alap>
    <id>529992</id>
    <szamlaszam>D-LOLO-66</szamlaszam>
    <tipus>D</tipus>
    <eszamla>0</eszamla>
    <kelt>2017-10-09</kelt>
    <telj>2017-10-09</telj>
    <fizh>2017-10-09</fizh>
    <fizmod>credit_card</fizmod>
    <fizmodunified>egy&#233;b</fizmodunified>
    <nyelv>hu</nyelv>
    <devizanem>HUF</devizanem>
    <devizaarf>0</devizaarf>
    <megjegyzes></megjegyzes>
```

```

    <penzforg>false</penzforg>
    <kata>true</kata>
    <teszt>false</teszt>
</alap>
<vevo>
    <id>221216</id>
    <nev>Ez nem & fog menni</nev>
    <cim>
        <orszag>Magyarorsz&#225;g</orszag>
        <irsz>1324</irsz>
        <telepules>Vesz&#337;Teszt,</telepules>
        <cim>1234</cim>
    </cim>
    <adoszam></adoszam>
    <fokonyv>
        <vevo></vevo>
        <vevoazon></vevoazon>
    </fokonyv>
</vevo>
<tetelek>
    <tetel>
        <nev>Alma</nev>
        <mennyiseg>1</mennyiseg>
        <mennyisegiegyseg>db</mennyisegiegyseg>
        <nettoegysegar>380</nettoegysegar>
        <afakulcs>20</afakulcs>
        <netto>380</netto>
        <arresafaalap>0</arresafaalap>
        <afa>76</afa>
        <brutto>456</brutto>
        <megjegyzes>Alma megjegyzés</megjegyzes>
        <fokonyv>
            <arbevetel></arbevetel>
            <afa></afa>
            <gazdasagiesemeny></gazdasagiesemeny>
            <gazdasagiesemenyafa></gazdasagiesemenyafa>
        </fokonyv>
    </tetel>
</tetelek>
<osszegek>
    <afakulcsossz>
        <afakulcs>20</afakulcs>
        <netto>464</netto>
        <afa>93</afa>
        <brutto>557</brutto>
    </afakulcsossz>
    <totalossz>
        <netto>464</netto>
        <afa>93</afa>
        <brutto>557</brutto>
    </totalossz>
</osszegek>
    <pdf>BASE64 formátumban található itt a számla PDF</pdf>
</szamla>

```

6.4.2 The response complies with the following XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.szamlazz.hu/xmlszamlavalasz"
xmlns:tns="http://www.szamlazz.hu/xmlszamlavalasz" elementFormDefault="qualified">
  <complexType name="szamlavalasz">
    <sequence>
      <element name="sikeres" type="int" maxOccurs="1" minOccurs="1"/></element>
      <element name="hibakod" type="string" maxOccurs="1" minOccurs="0"/></element>
      <element name="hibauzenet" type="string" maxOccurs="1" minOccurs="0"/></element>
      <element name="szamlaszam" type="string" maxOccurs="1" minOccurs="0"/></element>
      <element name="szamlanetto" type="double" maxOccurs="1" minOccurs="0"/></element>
      <element name="szamlabrutto" type="double" maxOccurs="1" minOccurs="0"/></element>
      <element name="vevoifiokurl" type="string" maxOccurs="1" minOccurs="0"/></element>
      <element name="pdf" type="base64Binary" maxOccurs="1" minOccurs="0"/></element>
    </sequence>
  </complexType>

  <element name="xmlszamlavalasz" type="tns:szamlavalasz"/></element>
</schema>
```

7 Deleting a pro forma invoice

7.1 Post request data

7.1.1 multipart/form-data based post

```
<html>
<head><meta content="text/html; CHARSET=UTF-8"/></head>
<body>
  <form action="https://www.szamlazz.hu/szamla/" method="post" enctype="multipart/form-data">
    <input type="file" name="action-szamla_agent_dijbekero_torlese"/><br>
    <input type="submit" name="generate" value="Díjbekérő törlése"/>
  </form>
</body>
</html>
```

7.1.2 Regular HTTP Post based method

7.2 Example XML

```
<?xml version="1.0" encoding="UTF-8"?>
<xmlszamladbkdél xmlns="http://www.szamlazz.hu/xmlszamladbkdél"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.szamlazz.hu/xmlszamladbkdél
http://www.szamlazz.hu/docs/xsds/szamladbkdél/xmlszamladbkdél.xsd">
  <beallitasok>
    <felhasznalo>Kérem töltse ki!</felhasznalo>
    <jelszo>Kérem töltse ki!</jelszo>
  </beallitasok>
  <fejlec>
    <szamlaszam>D-43</szamlaszam>
  </fejlec>
</xmlszamladbkdél>
```

The following XML structure has to be used to delete a pro forma invoice with the given order number:

```
<?xml version="1.0" encoding="UTF-8"?>
<xmlszamladbkdél xmlns="http://www.szamlazz.hu/xmlszamladbkdél"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.szamlazz.hu/xmlszamladbkdél
http://www.szamlazz.hu/docs/xsds/szamladbkdél/xmlszamladbkdél.xsd">
  <beallitasok>
    <felhasznalo>Kérem töltse ki!</felhasznalo>
```

```
<jelszo>Kérem töltse ki!</jelszo>  
</beallitasok>  
<fejlec>  
  <rendelesszam>XXX</rendelesszam>  
</fejlec>  
</xmlszamladbkdel>
```

7.3 XSD scheme compliance

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.szamlazz.hu/xmlszamladbkdel"
xmlns:tns="http://www.szamlazz.hu/xmlszamladbkdel" elementFormDefault="qualified">
  <complexType name="beallitasokTipus">
    <sequence>
      <element name="felhasznalo" type="string" maxOccurs="1" minOccurs="1"></element>
      <element name="jelszo" type="string" maxOccurs="1" minOccurs="1"></element>
    </sequence>
  </complexType>
  <complexType name="fejlecTipus">
    <sequence>
      <element name="szamlaszam" type="string" maxOccurs="1" minOccurs="0"></element>
      <element name="rendelesszam" type="string" maxOccurs="1" minOccurs="0"></element>
    </sequence>
  </complexType>
  <element name="xmlszamladbkdel">
    <complexType>
      <sequence>
        <element name="beallitasok" type="tns:beallitasokTipus" maxOccurs="1" minOccurs="1"></element>
        <element name="fejlec" type="tns:fejlecTipus" maxOccurs="1" minOccurs="1"></element>
      </sequence>
    </complexType>
  </element>
</schema>
```

7.4 Expected results of deleting pro forma invoice

Díjbekérő törlése során XML válasz érkezik. Kivéve kritikus hiba esetén, ilyenkor szövegesen írjuk ki a hibaát text/html formátumban.

Deleting a pro forma invoice results in an XML response. In case there was a critical error, you will receive the error message as regular text/html.

Expected XML response in case the deletion was succesful:

```
<?xml version="1.0" encoding="UTF-8"?>
<xmlszamladbkdelvalasz xmlns="http://www.szamlazz.hu/xmlszamladbkdelvalasz"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.szamlazz.hu/xmlszamladbkdelvalasz
http://www.szamlazz.hu/docs/xsds/szamladbkdel/xmlszamladbkdelvalasz.xsd ">
  <sikeres>true</sikeres>
</xmlszamladbkdelvalasz>
```

Expected XML response in case the deletion was unsuccessful:

```
<?xml version="1.0" encoding="UTF-8"?>
<xmlszamladbkdelvalasz xmlns="http://www.szamlazz.hu/xmlszamladbkdelvalasz"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.szamlazz.hu/xmlszamladbkdelvalasz
http://www.szamlazz.hu/docs/xsds/szamladbkdel/xmlszamladbkdelvalasz.xsd ">
  <sikeres>false</sikeres>
  <hibakod>335</hibakod>
  <hibauzenet><![CDATA[nincs ilyen díjbekérő a rendszerben, már törlésre került vagy nem is létezett]]></hibauzenet>
</xmlszamladbkdelvalasz>
```

7.5 XML response scheme

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
  xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.szamlazz.hu/xmlszamladbkdvalasz"
  xmlns:tns="http://www.szamlazz.hu/xmlszamladbkdvalasz"
  elementFormDefault="qualified">

  <element name="xmlszamladbkdvalasz">
    <complexType>
      <sequence>
        <element name="sikeres" type="boolean" maxOccurs="1" minOccurs="1"></element>
        <element name="hibakod" type="int" maxOccurs="1" minOccurs="0"></element>
        <element name="hibauzenet" type="string" maxOccurs="1" minOccurs="0"></element>
      </sequence>
    </complexType>
  </element>
</schema>
```

8 Generating a receipt

8.1 Post request data

8.1.1 multipart/form-data based post

```
<html>
<head><meta content="text/html; CHARSET=UTF-8"></head>
<body>
  <form action="https://www.szamlazz.hu/szamlal/" method="post" enctype="multipart/form-data">
    <input type="file" name="action-szaml_agent_nyugta_create"><br>
    <input type="submit" name="generate" value="Nyugta készítése">
  </form>
</body>
</html>
```

8.1.2 Regular HTTP Post based method

8.2 Example XML

```
<?xml version="1.0" encoding="UTF-8"?>
<xmInyugtacreate xmlns="http://www.szamlazz.hu/xmInyugtacreate" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.szamlazz.hu/xmInyugtacreate http://www.szamlazz.hu/docs/xsds/nyugta/xmInyugtacreate.xsd">

  <beallitasok>
    <felhasznalo>Kérem töltse ki!</felhasznalo>
    <jelszo>Kérem töltse ki!</jelszo>
    <pdfLetoltes>false</pdfLetoltes>
  </beallitasok>

  <fejlec>
    <hivasAzonosito></hivasAzonosito>
    <elotag>NYGTA</elotag>
    <fizmod>készpénz</fizmod>
  </fejlec>

  <penznem>Ft</penznem>
  <devizabank>MNB</devizabank>
  <devizaarf>0.0</devizaarf>
  <megjegyzes></megjegyzes>

  <!-- REQ -->
  <!-- REQ string --> <!-- a felhasználó loginneve -->
  <!-- REQ string --> <!-- a felhasználó jelszva -->
  <!-- REQ boolean --> <!-- PDF formátum letöltése -->

  <!-- REQ -->
  <!-- string --> <!-- a létrehozás egyedi azonosítója, duplikáció elkerülése -->
  <!-- REQ string --> <!-- nyugtaszám előtag, kötelező ==> NYGTA-2017-111 -->
  <!-- REQ string --> <!-- fizetési mód. bármilyen szöveg lehet, a felületen használt
értékek: átutalás, készpénz, bankkártya, csekk, utánvét, ajándékutalvány, barion, barter, csoportos beszédés, OTP Simple,
kompenzáció, kupon, PayPal,PayU, SZÉP kártya, utalvány -->
  <!-- REQ string --> <!-- pénznem: Ft, HUF, EUR, USD stb. -->
  <!-- string --> <!-- deviza (nem Ft/HUF) pénznem esetén az árfolyamot jegyző bank
neve -->
  <!-- string --> <!-- devizaárfolyam -->
  <!-- string --> <!-- általános szöveges megjegyzés, nyugtán megjelenik -->
```

```

<pdfSablon></pdfSablon>          <!-- string --> <!-- egyedi PDF sablon esetén annak azonosítója -->
<fokonyvVevo></fokonyvVevo>      <!-- string --> <!-- vevő főkönyvi azonosítója -->
</fejlec>

<tetelek>                          <!-- REQ -->
<tetel>                            <!-- REQ --> <!-- legalább egy tételre szükség van a nyugta kiállításához -->
  <megnevezes>Kiscicás lábtörlő</megnevezes> <!-- REQ string --> <!-- nyugta megnevezése -->
  <azonosito></azonosito> <!-- string --> <!-- nyugta azonosítója -->
  <mennyiseg>2.0</mennyiseg> <!-- REQ double --> <!-- tétel mennyisége -->
  <mennyisegiEgyseg>db</mennyisegiEgyseg> <!-- REQ string --> <!-- tétel mennyiségi egysége -->
  <nettoEgysegar>10000</nettoEgysegar> <!-- REQ double --> <!-- nettó egységár -->
  <netto>20000.0</netto> <!-- REQ double --> <!-- nettó érték (mennyiség * nettó egységár) -->
  <afakulcs>27</afakulcs> <!-- REQ string --> <!-- áfakulcs, értékek: 0, 5, 10, 27, AAM, TAM, EU, EUK, MAA,
F.AFA, K.AFA, ÁKK -->
  <afa>5400.0</afa> <!-- REQ double --> <!-- áfa összérték -->
  <brutto>25400.0</brutto> <!-- REQ double --> <!-- bruttó összérték -->
  <fokonyv> <!-- --> <!-- főkönyvi információ -->
    <arbevetel>...</arbevetel> <!-- string --> <!-- árbevétel főkönyvi szám -->
    <afa>...</afa> <!-- string --> <!-- áfa főkönyvi szám -->
  </fokonyv>
</tetel>
<tetel>
  <megnevezes>Kiskutyás lábtörlő</megnevezes>
  <mennyiseg>2.0</mennyiseg>
  <mennyisegiEgyseg>db</mennyisegiEgyseg>
  <nettoEgysegar>10000</nettoEgysegar>
  <netto>20000.0</netto>
  <afakulcs>ÁKK</afakulcs>
  <afa>5400.0</afa>
  <brutto>25400.0</brutto>
</tetel>
</tetelek>

<!--
A kifizetések nem kötelező, de ha meg van adva,
akkor az összegeknek meg kell egyezniük a számla végösszegével.
-->
<kifizetesek> <!-- string --> <!-- fizetési mód részletezése -->
<kifizetes>
  <fizetoeszkoz>utalvány</fizetoeszkoz> <!-- REQ string --> <!-- fizetőeszköz megnevezése -->
  <osszeg>30000.0</osszeg> <!-- REQ double --> <!-- fizetőeszközzel kiegyenlített összeg -->
  <leiras>OTP SZÉP kártya</leiras> <!-- double --> <!-- fizetőeszköz egyedi leírás -->
</kifizetes>
<kifizetes>
  <fizetoeszkoz>bankkártya</fizetoeszkoz>
  <osszeg>20800.0</osszeg>
</kifizetes>
</kifizetesek>

</xmlNyugtaCreate>

```

8.3 XSD scheme compliance

```

<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.szamlazz.hu/xmlNyugtaCreate"
xmlns:tns="http://www.szamlazz.hu/xmlNyugtaCreate" elementFormDefault="qualified">

```

```

  <complexType name="beallitasokTipus">
    <all>
      <element name="felhasznalo" type="string" maxOccurs="1" minOccurs="1"></element>
      <element name="jelszo" type="string" maxOccurs="1" minOccurs="1"></element>
      <element name="pdfLetoltes" type="boolean" maxOccurs="1" minOccurs="1"></element>
    </all>
  </complexType>

  <complexType name="fejlecTipus">
    <all>
      <element name="hivasAzonosito" type="string" maxOccurs="1" minOccurs="0"></element>
      <element name="elotag" type="string" maxOccurs="1" minOccurs="1"></element>
      <element name="fizmod" type="string" maxOccurs="1" minOccurs="1"></element>
    </all>
  </complexType>

```

```

    <element name="penznem" type="string" maxOccurs="1" minOccurs="1"></element>
    <element name="devizaarf" type="double" maxOccurs="1" minOccurs="0"></element>
    <element name="devizabank" type="string" maxOccurs="1" minOccurs="0"></element>
    <element name="megjegyzes" type="string" maxOccurs="1" minOccurs="0"></element>
    <element name="pdfSablon" type="string" maxOccurs="1" minOccurs="0"></element>
    <element name="fokonyvVevo" type="string" maxOccurs="1" minOccurs="0"></element>
  </all>
</complexType>

<complexType name="tetelTipus">
  <all>
    <element name="megnevezes" type="string" maxOccurs="1" minOccurs="1"></element>
    <element name="azonosito" type="string" maxOccurs="1" minOccurs="0"></element>
    <element name="mennysiseg" type="double" maxOccurs="1" minOccurs="1"></element>
    <element name="mennysisegEgyseg" type="string" maxOccurs="1" minOccurs="1"></element>
    <element name="nettoEgysegar" type="double" maxOccurs="1" minOccurs="1"></element>
    <element name="afakulcs" type="string" maxOccurs="1" minOccurs="1"></element>
    <element name="netto" type="double" maxOccurs="1" minOccurs="1"></element>
    <element name="afa" type="double" maxOccurs="1" minOccurs="1"></element>
    <element name="brutto" type="double" maxOccurs="1" minOccurs="1"></element>
    <element name="fokonyv" type="tns:tetelFokonyvTipus" maxOccurs="1" minOccurs="0"></element>
  </all>
</complexType>

<complexType name="tetelFokonyvTipus">
  <all>
    <element name="arbevétel" type="string" maxOccurs="1" minOccurs="0"></element>
    <element name="afa" type="string" maxOccurs="1" minOccurs="0"></element>
  </all>
</complexType>

<complexType name="tetelekTipus">
  <sequence>
    <element name="tetel" type="tns:tetelTipus" maxOccurs="unbounded" minOccurs="1"></element>
  </sequence>
</complexType>

<!-- jóváírások -->
<complexType name="kifizetesTipus">
  <all>
    <element name="fizetoeszköz" type="string" maxOccurs="1" minOccurs="1"></element>
    <element name="összeg" type="double" maxOccurs="1" minOccurs="1"></element>
    <element name="leiras" type="string" maxOccurs="1" minOccurs="0"></element>
  </all>
</complexType>

<complexType name="kifizetesekTipus">
  <sequence>
    <element name="kifizetes" type="tns:kifizetesTipus" maxOccurs="unbounded" minOccurs="1"></element>
  </sequence>
</complexType>

<element name="xmlNyugtaCreate">
  <complexType>
    <all>
      <element name="beallitasok" type="tns:beallitasokTipus" maxOccurs="1" minOccurs="1"></element>
      <element name="fejlec" type="tns:fejlecTipus" maxOccurs="1" minOccurs="1"></element>
      <element name="tetelek" type="tns:tetelekTipus" maxOccurs="1" minOccurs="1"></element>
      <element name="kifizetesek" type="tns:kifizetesekTipus" maxOccurs="1" minOccurs="0"></element>
    </all>
  </complexType>
</element>
</schema>

```

8.4 Expected results of generating receipts

The response contains everything about the generated receipt, and it can also include the PDF as well.

You should use `<hivasAzonosito />` in the posted XML to make the the call fault tolerant. If this field is in

use then it needs to be unique, otherwise the API call will be unsuccessful. This ensures that if the same XML is posted multiple times then it won't duplicate an existing receipt.

Sikeres nyugta minden esetben a < sikeres > true < / sikeres > mezőt tartalmazza. < sikeres > false < / sikeres > esetben a < hibakod / > és < hibauzenet / > mezők is kitöltésre kerülnek. A nyugta készítés során a számla agent részben ismertetett általános hibakódok mellett az alábbi hibakódok érkezhetnek a Számlázz.hu rendszerétől:

Every successful call contains the `<sikeres>true</sikeres>` field in the response. In case the receipt generation failed and the result field is false: `<sikeres>>false</sikeres>`, two additional fields will be included in the response, `<hibakod />` for the error code and `<hibauzenet />` for the error message itself. Apart from the general error messages described in the previous chapters, you may also receive one of the followings:

Error code	Description
336	"This prefix is already used to create invoices, and cannot be used to generate receipts." You need to use a prefix that is not already in use to create invoices. Please keep in mind that these prefixes that are used to create receipts are not visible on the frontend, which means that these prefixes used by the Invoice Agent will remain hidden.
337	Prefix is not in the expected format: it can only contain capital letters and numbers
338	The caller ID (unique field) you just used already exists on the server.
339	Receipt number does not exist (receipt query, send or delete).
340	The paid amount of the receipt differs from the gross amount of the receipt.

When you try to create a foreign currency receipt, and the currency field <penznem /> differs from either "Ft" or "HUF", then you also need to supply the exchange rate and the name of the bank providing the rates.

8.5 XML response example

```
<?xml version="1.0" encoding="UTF-8"?>
<xmlNyugtaValasz xmlns="http://www.szamlazz.hu/xmlNyugtaValasz" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.szamlazz.hu/xmlNyugtaValasz http://www.szamlazz.hu/docs/xsds/nyugta/xmlNyugtaValasz.xsd">
  <sikeres>true</sikeres>                                <!-- REQ boolean -->
  <hibakod></hibakod>                                     <!-- string --> <!-- hiba esetén a hibakód -->
  <hibauzenet></hibauzenet>                               <!-- string --> <!-- hiba esetén a hibauzenet -->

  <nyugtaPdf>...</nyugtaPdf>                            <!-- string --> <!-- nyugta PDF base64 enkódolással -->

  <nyugta>
    <alap>
      <id>123456</id>                                     <!-- REQ -->
      <hivasAzonosito></hivasAzonosito>                   <!-- REQ -->
      <nyugtaszam>NYGT-2017-123</nyugtaszam>              <!-- REQ id -->
      <tipus>NY</tipus>                                    <!-- string --> <!-- duplikáció elkerülése -->
    <!-- REQ string -->
    <!-- REQ string --> <!-- Nyugta típusa: NY esetén "Nyugta", SN esetén
"NYugta sztornó" -->
      <stornozott>false</stornozott>                      <!-- REQ string --> <!-- Nyugta típusa: NY esetén "Nyugta", SN esetén
"NYugta sztornó" -->
      <stornozottNyugtaszam>NYGT-2017-100</stornozottNyugtaszam> <!-- REQ boolean --> <!-- true ha a nyugta stornózott, false egyébként (NY
típus esetén érdekes) -->
      <stornozottNyugtaszam>NYGT-2017-100</stornozottNyugtaszam> <!-- string --> <!-- Ha a nyugta sztornózó, akkor
a sztornózott számla száma (SN típus esetén jelzi melyik nyugta lett sztornózva) -->
      <kelt>2015-12-01</kelt>                              <!-- REQ date --> <!-- Nyugta kelte -->
      <fizmod>készpénz</fizmod>                           <!-- REQ string -->
      <penznem>EUR</penznem>                               <!-- REQ string --> <!-- üres, Ft, HUF, EUR, CHF, USD, ... -->
      <devizabank></devizabank>                             <!-- string -->
      <devizaarf>210</devizaarf>                           <!-- double -->
      <megjegyzes></megjegyzes>                             <!-- string -->
      <fokonyvVevo></fokonyvVevo>                         <!-- string --> <!-- könyvelési információ -->
      <teszt>false</teszt>                                <!-- REQ boolean --> <!-- Teszt vagy valós céggel lett létrehozva a nyugta -->

```

```

</alap>

<tetelek>
  <tétel>
    <azonosito></azonosito>                                <!-- string --> <!-- tétel azonosítója -->
    <megnevezes>Kiscicás lábtörlő</megnevezes>           <!-- REQ string --> <!-- tétel megnevezése -->
    <mennyiseg>2.0</mennyiseg>                             <!-- REQ double --> <!-- tétel mennyisége -->
    <mennyisegiEgyseg>db</mennyisegiEgyseg>               <!-- REQ string --> <!-- tétel mennyiségi egysége -->
    <nettoEgysegar>10000</nettoEgysegar>                  <!-- REQ double --> <!-- tétel nettó egységár -->
    <netto>20000.0</netto>                                <!-- REQ double --> <!-- tétel nettó érték (egységár * mennyiség) -->
    <afakulcs>27</afakulcs>                               <!-- REQ double --> <!-- áfakulcs -->
    <afa>5400.0</afa>                                     <!-- REQ double --> <!-- áfa értéke -->
    <brutto>25400.0</brutto>                              <!-- REQ double --> <!-- bruttó érték -->
    <fokonyv>                                              <!-- --> <!-- tétel főkönyvi információk -->
      <arbevétel></arbevétel >                           <!-- string --> <!-- árbevétel főkönyvi száma -->
      <afa></afa>                                           <!-- string --> <!-- áfa főkönyvi száma -->
    </fokonyv>
  </tétel>
  <tétel>
    <megnevezes>Kiskutyás lábtörlő</megnevezes>
    <mennyiseg>2.0</mennyiseg>
    <mennyisegiEgyseg>db</mennyisegiEgyseg>
    <nettoEgysegar>10000</nettoEgysegar>
    <nettoErtek>20000.0</nettoErtek>
    <afakulcs>27</afakulcs>
    <afaErtek>5400.0</afaErtek>
    <bruttoErtek>25400.0</bruttoErtek>
  </tétel>
</tetelek>

<kifizetesek>
  <kifizetes>
    <fizetoeszkoz>utalvány</fizetoeszkoz>
    <osszeg>1000.0</osszeg>
    <leiras>OTP SZÉP kártya</leiras>
  </kifizetes>
  <kifizetes>
    <fizetoeszkoz>bankkártya</fizetoeszkoz>
    <osszeg>3000.0</osszeg>
  </kifizetes>
</kifizetesek>

<osszegek>
  <afakulcssossz>                                         <!-- REQ int -->
  <afatipus>ÁKK</afatipus>                               <!-- string --> <!-- ÁKK: áfakörön kívüli, K.AFA: különbszeti áfa,
F.AFA: fordított áfa fizetés, MAA: Mentés az adó alól, EUK: EU-n kívül, EU: EU-n belül, AAM: Alanyi adómentes, TAM: Tárgyi
adómentes -->
  <afakulcs>0</afakulcs>                                <!-- REQ double -->
  <netto>200</netto>                                     <!-- REQ double -->
  <afa>54</afa>                                           <!-- REQ double -->
  <brutto>254</brutto>                                   <!-- REQ double -->
  </afakulcssossz>
  <totalossz>                                             <!-- REQ -->
  <netto>200</netto>                                     <!-- REQ double -->
  <afa>54</afa>                                           <!-- REQ double -->
  <brutto>254</brutto>                                   <!-- REQ double -->
  </totalossz>
</osszegek>
</nyugta>

</xmlNyugtavalasz>

```

8.6 XML response scheme

```

<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.szamlazz.hu/xmlNyugtavalasz"
xmlns:tns="http://www.szamlazz.hu/xmlNyugtavalasz" elementFormDefault="qualified">

  <simpleType name="afatipusTipus">
    <restriction base="string">

```

```

    <enumeration value="TAM"></enumeration>
    <enumeration value="AAM"></enumeration>
    <enumeration value="EU"></enumeration>
    <enumeration value="EUK"></enumeration>
    <enumeration value="MAA"></enumeration>
    <enumeration value="F.AFA"></enumeration>
    <enumeration value="K.AFA"></enumeration>
    <enumeration value="ÁKK"></enumeration>
  </restriction>
</simpleType>

<simpleType name="nyugtatipusTipus">
  <restriction base="string">
    <enumeration value="NY"></enumeration>      <!-- nyugta -->
    <enumeration value="SN"></enumeration>      <!-- nyugta sztorító -->
  </restriction>
</simpleType>

<!-- számla alapadatai -->
<complexType name="alapTipus">
  <all>
    <element name="id" type="int" maxOccurs="1" minOccurs="1"></element>
    <element name="hivasAzonosito" type="string" maxOccurs="1" minOccurs="0"></element>
    <element name="nyugtaszam" type="string" maxOccurs="1" minOccurs="1"></element>
    <element name="tipus" type="tns:nyugtatipusTipus" maxOccurs="1" minOccurs="1"></element>
    <element name="stornozott" type="boolean" maxOccurs="1" minOccurs="1"></element>
    <element name="stornozottNyugtaszam" type="string" maxOccurs="1" minOccurs="0"></element>
    <element name="kelt" type="date" maxOccurs="1" minOccurs="1"></element>
    <element name="fizmod" type="string" maxOccurs="1" minOccurs="1"></element>
    <element name="penznem" type="string" maxOccurs="1" minOccurs="1"></element>
    <element name="devizabank" type="string" maxOccurs="1" minOccurs="0"></element>
    <element name="devizaarf" type="double" maxOccurs="1" minOccurs="0"></element>
    <element name="megjegyzes" type="string" maxOccurs="1" minOccurs="0"></element>
    <element name="fokonyvVevo" type="string" maxOccurs="1" minOccurs="0"></element>
    <element name="teszt" type="boolean" maxOccurs="1" minOccurs="1"></element>
  </all>
</complexType>

<!-- számla tételek -->
<complexType name="fokonyvtetelTipus">
  <all>
    <element name="arbevetel" type="string" maxOccurs="1" minOccurs="0"></element>
    <element name="afa" type="string" maxOccurs="1" minOccurs="0"></element>
  </all>
</complexType>

<complexType name="tetelTipus">
  <all>
    <element name="megnevezes" type="string" maxOccurs="1" minOccurs="1"></element>
    <element name="azonosito" type="string" maxOccurs="1" minOccurs="0"></element>
    <element name="nettoEgysegar" type="double" maxOccurs="1" minOccurs="1"></element>
    <element name="mennyiseg" type="double" maxOccurs="1" minOccurs="1"></element>
    <element name="mennyisegiEgyseg" type="string" maxOccurs="1" minOccurs="1"></element>
    <element name="netto" type="double" maxOccurs="1" minOccurs="1"></element>
    <element name="afatipus" type="tns:afatipusTipus" maxOccurs="1" minOccurs="0"></element>
    <element name="afakulcs" type="int" maxOccurs="1" minOccurs="1"></element>
    <simpleType>
      <restriction base="int">
        <minInclusive value="0"></minInclusive>
      </restriction>
    </simpleType>
  </all>
  <all>
    <element name="afa" type="double" maxOccurs="1" minOccurs="1"></element>
    <element name="brutto" type="double" maxOccurs="1" minOccurs="1"></element>
    <element name="fokonyv" type="tns:fokonyvtetelTipus" maxOccurs="1" minOccurs="0"></element>
  </all>
</complexType>

<complexType name="tetelFokonyvTipus">
  <all>
    <element name="arbevetelFokonyviSzam" type="string" maxOccurs="1" minOccurs="0"></element>
    <element name="afaFokonyviSzam" type="string" maxOccurs="1" minOccurs="0"></element>
  </all>
</complexType>

```

```

    </all>
</complexType>
<complexType name="tetelekTipus">
    <sequence>
        <element name="tetel" type="tns:tetelTipus" maxOccurs="unbounded" minOccurs="1"></element>
    </sequence>
</complexType>

<!-- áfakulcsonkénti összesítés -->
<complexType name="afakulcsosszTipus">
    <all>
        <element name="afatipus" type="tns:afatipusTipus" maxOccurs="1" minOccurs="0"></element>
        <element name="afakulcs" maxOccurs="1" minOccurs="1">
            <simpleType>
                <restriction base="int">
                    <minInclusive value="0"></minInclusive>
                </restriction>
            </simpleType>
        </element>
        <element name="netto" type="double" maxOccurs="1" minOccurs="1"></element>
        <element name="afa" type="double" maxOccurs="1" minOccurs="1"></element>
        <element name="brutto" type="double" maxOccurs="1" minOccurs="1"></element>
    </all>
</complexType>
<!-- számla teljes összesítés -->
<complexType name="totalosszTipus">
    <all>
        <element name="netto" type="double" maxOccurs="1" minOccurs="1"></element>
        <element name="afa" type="double" maxOccurs="1" minOccurs="1"></element>
        <element name="brutto" type="double" maxOccurs="1" minOccurs="1"></element>
    </all>
</complexType>
<!-- számla összesítők (áfakulcsonként és total) -->
<complexType name="osszegekTipus">
    <sequence>
        <element name="afakulcsossz" type="tns:afakulcsosszTipus" maxOccurs="unbounded" minOccurs="1"></element>
        <element name="totalossz" type="tns:totalosszTipus" maxOccurs="1" minOccurs="1"></element>
    </sequence>
</complexType>

<!-- jóváírások -->
<complexType name="kifizetesTipus">
    <all>
        <element name="fizetoeszkoz" type="string" maxOccurs="1" minOccurs="1"></element>
        <element name="osszeg" type="double" maxOccurs="1" minOccurs="1"></element>
        <element name="leiras" type="string" maxOccurs="1" minOccurs="0"></element>
    </all>
</complexType>
<complexType name="kifizetesekTipus">
    <sequence>
        <element name="kifizetes" type="tns:kifizetesTipus" maxOccurs="unbounded" minOccurs="1"></element>
    </sequence>
</complexType>

<complexType name="nyugtaTipus">
    <all>
        <element name="alap" type="tns:alapTipus" maxOccurs="1" minOccurs="1"></element>
        <element name="tetelek" type="tns:tetelekTipus" maxOccurs="1" minOccurs="1"></element>
        <element name="kifizetesek" type="tns:kifizetesekTipus" maxOccurs="1" minOccurs="0"></element>
        <element name="osszegek" type="tns:osszegekTipus" maxOccurs="1" minOccurs="1"></element>
    </all>
</complexType>

<element name="xmlNyugtavalasz">
    <complexType>
        <all>
            <element name="sikeres" type="boolean" maxOccurs="1" minOccurs="1"></element>
            <element name="hibakod" type="int" maxOccurs="1" minOccurs="0"></element>
            <element name="hibauzenet" type="string" maxOccurs="1" minOccurs="0"></element>
            <element name="nyugtaPdf" type="string" maxOccurs="1" minOccurs="0"></element>
            <element name="nyugta" type="tns:nyugtaTipus" maxOccurs="1" minOccurs="0"></element>
        </all>
    </complexType>

```

```

    </all>
  </complexType>
</element>
</schema>

```

9 Reversing a receipt (storno)

9.1 Post request data

9.1.1 multipart/form-data based post

```

<html>
  <head><meta content="text/html; CHARSET=UTF-8"></head>
  <body>
    <form action="https://www.szamlazz.hu/szamla/" method="post" enctype="multipart/form-data">
      <input type="file" name="action-szamla_agent_nyugta_storno"><br>
      <input type="submit" name="generate" value="Nyugta sztornózása">
    </form>
  </body>
</html>

```

9.1.2 Regular HTTP Post based method

9.2 Example XML

```

<?xml version="1.0" encoding="UTF-8"?>
<xmlNyugtas xmlns="http://www.szamlazz.hu/xmlNyugtas" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.szamlazz.hu/xmlNyugtas http://www.szamlazz.hu/docs/xsds/nyugtas/xmlNyugtas.xsd">
  <beallitasok>
    <felhasznalo>Kérem töltse ki!</felhasznalo>
    <jelszo>Kérem töltse ki!</jelszo>
    <pdfLetoltes>>false</pdfLetoltes>      <!-- PDF számla letöltése -->
  </beallitasok>
  <fejlec>
    <nyugtaszam>NYGT-2020-1</nyugtaszam>    <!-- stornózandó számla száma -->

    <!-- Egyedi PDF sablon (nyugtakép) azonosítója -->
    <!-- <pdfSablon></pdfSablon> -->
  </fejlec>
</xmlNyugtas>

```

9.3 XSD scheme compliance

```

<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.szamlazz.hu/xmlNyugtas"
xmlns:tns="http://www.szamlazz.hu/xmlNyugtas" elementFormDefault="qualified">

  <complexType name="beallitasokTipus">
    <sequence>
      <element name="felhasznalo" type="string" maxOccurs="1" minOccurs="1"></element>
      <element name="jelszo" type="string" maxOccurs="1" minOccurs="1"></element>
      <element name="pdfLetoltes" type="boolean" maxOccurs="1" minOccurs="1"></element>
    </sequence>
  </complexType>

  <complexType name="fejlecTipus">
    <sequence>
      <element name="nyugtaszam" type="string" maxOccurs="1" minOccurs="1"></element>
      <element name="pdfSablon" type="string" maxOccurs="1" minOccurs="0"></element>
      <element name="hivasAzonosito" type="string" maxOccurs="1" minOccurs="0"></element>
    </sequence>

```

```

</complexType>

<element name="xmlNyugta">
  <complexType>
    <sequence>
      <element name="beallitasok" type="tns:beallitasokTipus" maxOccurs="1" minOccurs="1"></element>
      <element name="fejlec" type="tns:fejlecTipus" maxOccurs="1" minOccurs="1"></element>
    </sequence>
  </complexType>
</element>

</schema>

```

9.4 Expected results of reversing a receipts

The expected result matches the response when generating new receipts.

10 Querying a receipt

10.1 Post request data

10.1.1 multipart/form-data based post

```

<html>
<head><meta content="text/html; CHARSET=UTF-8"></head>
<body>
  <form action="https://www.szamlazz.hu/szamla/" method="post" enctype="multipart/form-data">
    <input type="file" name="action-szamla_agent_nyugta_get"><br>
    <input type="submit" name="generate" value="Nyugta lekérdezése">
  </form>
</body>
</html>

```

10.1.2 Regular HTTP Post based method

10.2 Example XML

```

<?xml version="1.0" encoding="UTF-8"?>
<xmlNyugtaget xmlns="http://www.szamlazz.hu/xmlNyugtaget" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.szamlazz.hu/xmlNyugtaget http://www.szamlazz.hu/docs/xsds/nyugtaget/xmlNyugtaget.xsd">
  <beallitasok>
    <felhasznalo>Kérem töltse ki!</felhasznalo>      <!-- REQ string --> <!-- felhasználó loginneve -->
    <jelszo>Kérem töltse ki!</jelszo>                <!-- REQ string --> <!-- felhasználó jelszava -->
    <pdfLetoltes>false</pdfLetoltes>                <!-- REQ string --> <!-- PDF formátum letöltése -->
  </beallitasok>
  <fejlec>
    <nyugtaszam>NYGT-2020-1</nyugtaszam>            <!-- REQ string --> <!-- a letölteni kívánt számla azonosítója -->

    <!-- Egyedi PDF sablon (nyugtakép) azonosítója -->
    <!-- <pdfSablon></pdfSablon> -->
  </fejlec>
</xmlNyugtaget>

```

10.3 XSD scheme compliance

```

<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.szamlazz.hu/xmlNyugtaget"
xmlns:tns="http://www.szamlazz.hu/xmlNyugtaget" elementFormDefault="qualified">

  <complexType name="beallitasokTipus">
    <all>
      <element name="felhasznalo" type="string" maxOccurs="1" minOccurs="1"></element>

```

```

    <element name="jelszo" type="string" maxOccurs="1" minOccurs="1"></element>
    <element name="pdfLetoltes" type="boolean" maxOccurs="1" minOccurs="1"></element>
  </all>
</complexType>

<complexType name="fejlecTipus">
  <all>
    <element name="nyugtaszam" type="string" maxOccurs="1" minOccurs="1"></element>
    <element name="pdfSablon" type="string" maxOccurs="1" minOccurs="0"></element>
  </all>
</complexType>

<element name="xmlNyugtaget">
  <complexType>
    <all>
      <element name="beallitasok" type="tns:beallitasokTipus" maxOccurs="1" minOccurs="1"></element>
      <element name="fejlec" type="tns:fejlecTipus" maxOccurs="1" minOccurs="1"></element>
    </all>
  </complexType>
</element>
</schema>

```

10.4 Expected results of querying a receipt

The expected result matches the response when generating new receipts.

11 Sending a receipt

11.1 Post request data

11.1.1 multipart/form-data based post

```

<html>
<head><meta content="text/html; CHARSET=UTF-8"></head>
<body>
  <form action="https://www.szamlazz.hu/szamlaz" method="post" enctype="multipart/form-data">
    <input type="file" name="action-szamlaz_agent_nyugta_send"><br>
    <input type="submit" name="generate" value="Nyugta kiküldése">
  </form>
</body>
</html>

```

11.1.2 Regular HTTP Post based method

11.2 Example XML

```

<?xml version="1.0" encoding="UTF-8"?>
<xmlNyugtasend xmlns="http://www.szamlazz.hu/xmlnyugtasend" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.szamlazz.hu/xmlnyugtasend
http://www.szamlazz.hu/docs/xsds/nyugtasend/xmlnyugtasend.xsd">

```

```

  <beallitasok>
    <felhasznalo>Kérem töltse ki!</felhasznalo>
    <jelszo>Kérem töltse ki!</jelszo>
  </beallitasok>

  <fejlec>
    <!-- csak már elkészített számlához lehet e-mail értesítőt küldeni -->
    <nyugtaszam>NYGT-2020-1</nyugtaszam>
  </fejlec>

  <!-- e-mail tartalma -->

```

```

<emailKuldes>
  <!-- e-mail adatai, ha nincs megadva akkor a korábbi e-mail lesz kiküldve -->
  <email>vevo@example.com</email>           <!-- string --> <!-- vevő e-mail címe (ide megy ki a levél) -->
  <emailReplyto>elado@example.com</emailReplyto> <!-- string --> <!-- reply-to cím, erre a címre válaszol a vevő -->
  <emailTargy>Email tárgya</emailTargy>       <!-- string --> <!-- e-mail tárgya -->
  <emailSzoveg>                               <!-- string --> <!-- e-mail szövege -->

```

Ez az email szövege.

Aláírás

```

</emailSzoveg>
</emailKuldes>
</xmlNyugtasend>

```

11.3 XSD scheme compliance

```

<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.szamlazz.hu/xmlNyugtasend"
xmlns:tns="http://www.szamlazz.hu/xmlNyugtasend" elementFormDefault="qualified">

```

```

  <complexType name="beallitasokTipus">
    <sequence>
      <element name="felhasznalo" type="string" maxOccurs="1" minOccurs="1"></element>
      <element name="jelszo" type="string" maxOccurs="1" minOccurs="1"></element>
    </sequence>
  </complexType>

  <complexType name="fejlecTipus">
    <sequence>
      <element name="nyugtaszam" type="string" maxOccurs="1" minOccurs="1"></element>
    </sequence>
  </complexType>

  <complexType name="emailKuldes">
    <sequence>
      <element name="email" type="string" maxOccurs="1" minOccurs="0"></element>
      <element name="emailReplyto" type="string" maxOccurs="1" minOccurs="0"></element>
      <element name="emailTargy" type="string" maxOccurs="1" minOccurs="0"></element>
      <element name="emailSzoveg" type="string" maxOccurs="1" minOccurs="0"></element>
    </sequence>
  </complexType>

  <element name="xmlNyugtasend">
    <complexType>
      <sequence>
        <element name="beallitasok" type="tns:beallitasokTipus" maxOccurs="1" minOccurs="1"></element>
        <element name="fejlec" type="tns:fejlecTipus" maxOccurs="1" minOccurs="1"></element>
        <element name="emailKuldes" type="tns:emailKuldes" maxOccurs="1" minOccurs="0"></element>
      </sequence>
    </complexType>
  </element>
</schema>

```

11.4 Expected results of sending a receipt

Whether sending the receipt was successful or not depends on received response:

<sikeres>true</sikeres> or <sikeres>false</sikeres>. If the sending was unsuccessful, you will receive an error code and one of the following error messages.

11.4.1 Example response: success

```

<?xml version="1.0" encoding="UTF-8"?>
<xmlNyugtasendvalasz xmlns="http://www.szamlazz.hu/xmlNyugtasendvalasz"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.szamlazz.hu/xmlNyugtasendvalasz
http://www.szamlazz.hu/docs/xsds/nyugta/xmlNyugtasendvalasz.xsd">
  <sikeres>true</sikeres>
  <hibakod></hibakod>
  <hibauzenet></hibauzenet>
</xmlNyugtasendvalasz>

```

11.4.2 Example response: failed

```
<?xml version="1.0" encoding="UTF-8"?>
<xmlNyugtasendvalasz xmlns="http://www.szamlazz.hu/xmlNyugtasendvalasz"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.szamlazz.hu/xmlNyugtasendvalasz
http://www.szamlazz.hu/docs/xsds/nyugta/xmlNyugtasendvalasz.xsd">
  <sikeres>false</sikeres>
  <hibakod>7</hibakod>
  <hibauzenet>Hiányzó adat: emailtargy elem.</hibauzenet>
</xmlNyugtasendvalasz>
```

11.5 XML response scheme

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.szamlazz.hu/xmlNyugtasendvalasz"
xmlns:tns="http://www.szamlazz.hu/xmlNyugtasendvalasz" elementFormDefault="qualified">

  <element name="xmlNyugtasendvalasz">
    <complexType>
      <all>
        <element name="sikeres" type="boolean" maxOccurs="1" minOccurs="1"></element>
        <element name="hibakod" type="int" maxOccurs="1" minOccurs="0"></element>
        <element name="hibauzenet" type="string" maxOccurs="1" minOccurs="0"></element>
      </all>
    </complexType>
  </element>
</schema>
```

12 Querying taxpayers

The interface is for querying the validity of a VAT number. The data is from the Hungarian National Tax and Customs Administration's Online Invoice Platform.

13.1 Post request data

13.1.1 multipart/form-data based post

```
<html>
<head><meta content="text/html; CHARSET=UTF-8"></head>
<body>
  <form action="https://www.szamlazz.hu/szamlaz/" method="post" enctype="multipart/form-data">
    <input type="file" name="action-szamlaz_agent_taxpayer"><br>
    <input type="submit" name="generate" value="Adózó lekérdezése">
  </form>
</body>
</html>
```

13.1.2 Regular HTTP Post based method

13.2 Example XML

```
<?xml version="1.0" encoding="UTF-8"?>
<xmltaxpayer xmlns="http://www.szamlazz.hu/xmltaxpayer"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.szamlazz.hu/xmltaxpayerhttp://www.szamlazz.hu/docs/xsds/agent/xmltaxpayer.xsd">
  <beallitasok>
    <felhasznalo>Kérem töltse ki!</felhasznalo>
    <jelszo>Kérem töltse ki!</jelszo>
  </beallitasok>
```

```

    <torzsszam>12345678</torzsszam>
</xmntaxpayer>

```

13.3 XSD scheme compliance

```

<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.szamlazz.hu/xmntaxpayer"
  xmlns:tns="http://www.szamlazz.hu/xmntaxpayer"
  elementFormDefault="qualified">
  <complexType name="beallitasokTipus">
    <sequence>
      <element name="felhasznalo" type="string" maxOccurs="1"
        minOccurs="1"></element>
      <element name="jelszo" type="string" maxOccurs="1"
        minOccurs="1"></element>
    </sequence>
  </complexType>
  <simpleType name="torszsszamTipus">
    <restriction base="string">
      <length value="8" />
      <pattern value="[0-9]{8}" />
    </restriction>
  </simpleType>
  <element name="xmntaxpayer">
    <complexType>
      <sequence>
        <element name="beallitasok" type="tns:beallitasokTipus"
          maxOccurs="1" minOccurs="1"></element>
        <element name="torzsszam" type="tns:torszsszamTipus"
          maxOccurs="1" minOccurs="1"></element>
      </sequence>
    </complexType>
  </element>
</schema>

```

13.4 Expected results of querying a taxpayer

The response is always matches the Hungarian National Tax and Customs Administration’s Online Invoice Platform’s QueryTaxPayerResponse type.

13.4.1 Example response: success

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<QueryTaxpayerResponse
  xmlns="http://schemas.nav.gov.hu/OSA/1.0/api"
  xmlns:ns2="http://schemas.nav.gov.hu/OSA/1.0/data">
  <header>
    <requestId>170_k8knfskxnjdn24p97errf7</requestId>
    <timestamp>2018-07-05T10:57:46.810Z</timestamp>
    <requestVersion>1.0</requestVersion>
  </header>
  <result>
    <funcCode>OK</funcCode>
  </result>
  <taxpayerValidity>true</taxpayerValidity>
  <taxpayerData>
    <taxpayerName>KBOSS.HU KERESKEDELMI ÉS SZOLGÁLTATÓ KORLÁTOLT
      FELELŐSSÉGŰ TÁRSASÁG</taxpayerName>
    <taxpayerAddress>
      <countryCode>HU</countryCode>
      <postalCode>1031</postalCode>
      <city>BUDAPEST</city>
      <streetName>ZÁHONY</streetName>
    </taxpayerAddress>
  </taxpayerData>
</QueryTaxpayerResponse>

```

```

                <publicPlaceCategory>UTCA</publicPlaceCategory>
                <number>7.</number>
            </taxpayerAddress>
        </taxpayerData>
    </QueryTaxpayerResponse>

```

or

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<QueryTaxpayerResponse
  xmlns="http://schemas.nav.gov.hu/OSA/1.0/api"
  xmlns:ns2="http://schemas.nav.gov.hu/OSA/1.0/data">
  <header>
    <requestId>170_9ugu5pcfyyiu8kdsimcfyi</requestId>
    <timestamp>2018-07-05T11:01:56.582Z</timestamp>
    <requestVersion>1.0</requestVersion>
  </header>
  <result>
    <funcCode>OK</funcCode>
  </result>
  <taxpayerValidity>>false</taxpayerValidity>
</QueryTaxpayerResponse>

```

13.4.2 Example response: failed

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<QueryTaxpayerResponse
  xmlns="http://schemas.nav.gov.hu/OSA/1.0/api"
  xmlns:ns2="http://schemas.nav.gov.hu/OSA/1.0/data">
  <header>
    <requestId>170_eiq6ffnzhcyrwc7w5qj84</requestId>
    <timestamp>2018-07-05T11:03:54.326Z</timestamp>
    <requestVersion>1.0</requestVersion>
  </header>
  <result>
    <funcCode>ERROR</funcCode>
    <errorCode>374</errorCode>
    <message>A NAV rendszere szerint: Helytelen kérés!</message>
  </result>
</QueryTaxpayerResponse>

```

13.5 XML response scheme

The response scheme is in the Hungarian National Tax and Customs Administration's Online Invoice Platform's documentation.