

## PFR Agent – technical documentation and instructions

### v1.10

1.	Activating the PFR Agent module.....	3
2.	Configuring PFR Agent.....	4
2.1	Configuring file sharing .....	4
2.2	Configuring invoice layout.....	6
2.3	Configure invoice printing .....	7
3.	Data exchange via file system .....	8
4.	Examples of commands in XML format.....	11
4.1	Attention .....	11
4.2	VerifyPin .....	11
4.3	CreateInvoice.....	11
4.4	GetLastSignedInvoice .....	15
4.5	GetStatus .....	15
4.6	GetEnvironmentParameters .....	21
4.7	GetSubject .....	21
5.	Intermediary web services .....	22
5.1	Creating an invoice.....	22
5.2	Getting the last signed invoice .....	22
5.3	Providing an image of a previously signed invoice .....	22
6.	Model extension.....	22
7.	Encoding and tax rates .....	24
8.	Reports .....	26
8.1	Web services for generating and printing reports .....	27
8.2	File command system for generating and printing reports .....	28
8.3	Response structure (web services and command file system) .....	30



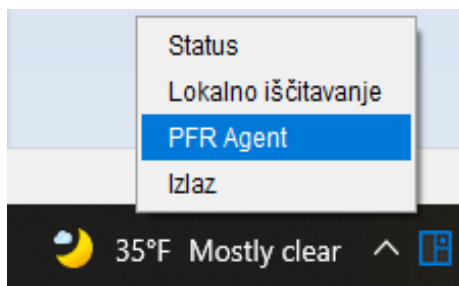
## 1. Activating the PFR Agent module

**PFR Agent** is a module of the myLPFR application and represents middleware, ie. intermediary service, which aims to assist ESIR suppliers in fiscalizing invoices. It is intended primarily for solutions developed in technologies that have limited communication via HTTP transport. PFR Agent provides the ability to communicate with PFR services through the file system, or through messaging through XML and JSON files. In addition to this basic function, all ESIR solutions also get the ability to print a fiscalized invoice on any thermal printer that supports the ESC / POS protocol.

*PFR Agent* functions as an independent process within the same container in which the myLPFR application is running and, if activated, runs along with it.

**Process activation must first be enabled through the "application.properties" configuration file located at the location where myLPFR is installed, in the "config" subfolder (eg C:\myLPFR\config\application.properties). In this file you need to change the line enable-agent = false and set the value to enable-agent = true.**

After launching the application, the option to configure the Agent will appear in the main menu, which is an indicator that the process has started successfully.



## 2. Configuring PFR Agent

### 2.1 Configuring file sharing

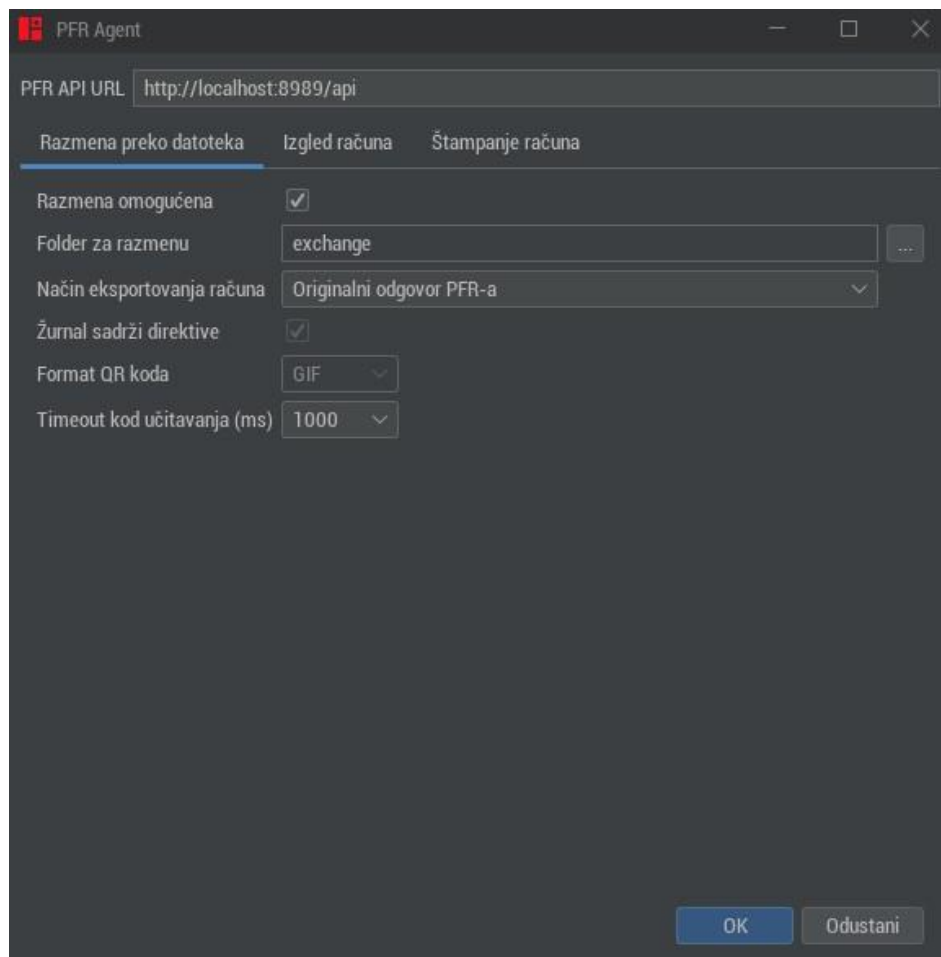
After you choose *PFR Agent* option from application main menu, configuration window will be shown.

In the first row inside the window, the user can enter the base address where the PFR API is located. If this field is left blank, the default address will be <http://localhost:8989/api>, which means that the PFR Agent will use myLPFR as its LPFR. This is the address that the PFR Agent will use to forward the commands received in the files when exchanging over the file system.

**IMPORTANT: For the purposes of certification of your ESIR, set the development L-PFR of the Tax Administration as the PFR API URL-**

[http://devesdc.sandbox.suf.purs.gov.rs:8888/VAŠ\\_TOKEN/api](http://devesdc.sandbox.suf.purs.gov.rs:8888/VAŠ_TOKEN/api)

Below are two sections through which the data exchange is configured, ie the printing of signed invoices.



In order for the exchange to work, it is necessary to select the "Razmena omogućena" option. It is then possible to configure several parameters related to the export of certain segments from the account signing results to separate files. You can currently select the following export methods:

- **Originalni odgovor PFR-a** - after signing the invoice, in the "from-sdc" folder you get a standard answer in JSON / XML format
- **Žurnal i QR kod** - after the response received from the PFR, the journal and QR code will be "unpacked" and saved as two separate files in the "from-sdc" folder. If an option "Žurnal sadrži direktive" is selected, the text file will contain a directive **{{qr-code}}** at the position where the QR code should be printed, while each line that needs to be printed in twice the font size will start with a sequence **!{{** , and end with a sequence **}}!**. In addition to this, it is possible to choose the file format to which the QR code will be exported.
- **Slika računa** - after receiving a response from PFR, a PNG image of the account is generated and saved in the "from-sdc" folder
- **Ne eksportuj ništa** - after signing the invoice, the answer will not be saved in the "from-sdc" folder. This option can be useful in case the PFR Agent is configured to print invoices after receiving a response from PFR. In this case, the "from-sdc" folder should be monitored only for possible errors that PFR can return.

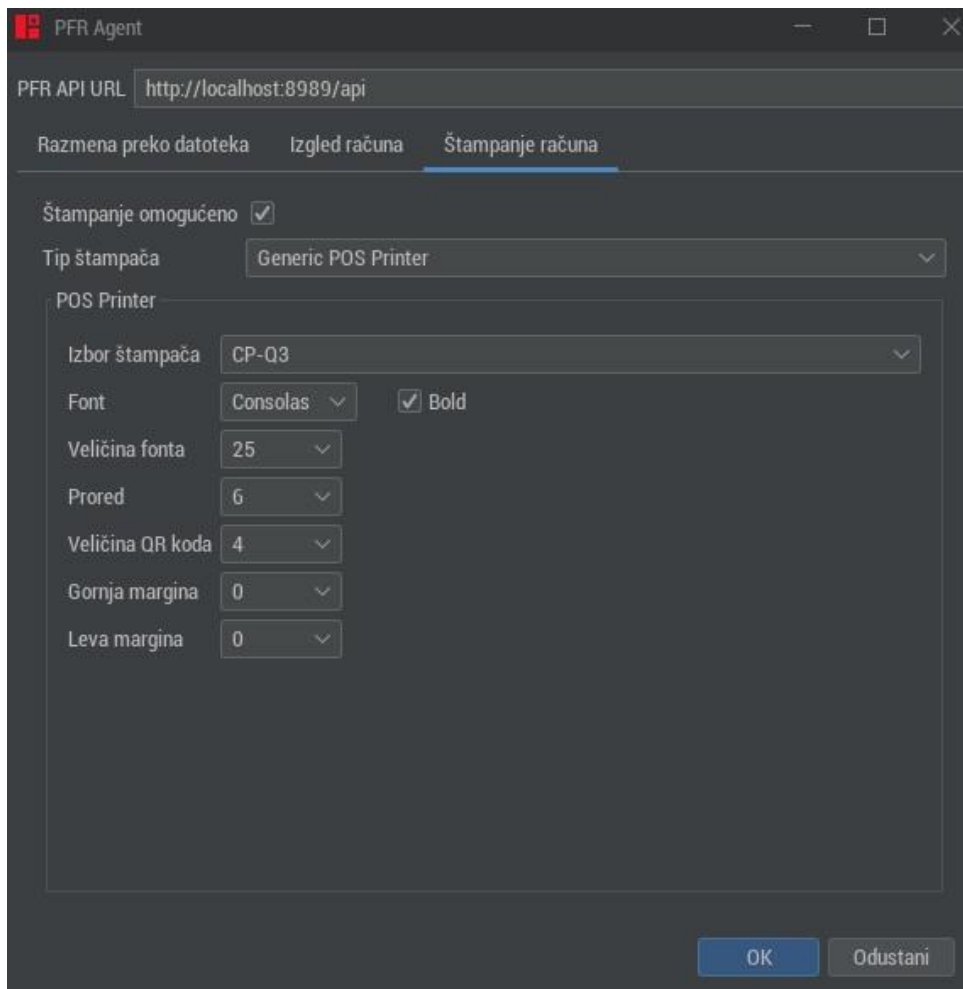
## 2.2 Configuring invoice layout

The screenshot shows the 'PFR Agent' configuration window. At the top, there's a 'PFR API URL' field with the value 'http://localhost:8989/api'. Below this are three tabs: 'Razmena preko datoteka', 'Izgled računa' (which is selected), and 'Štampanje računa'. Under the 'Izgled računa' tab, there are three sections: 'Isečak' (Snippet), 'A4', and 'Logotip'. The 'Isečak' section has a 'Dužina reda' (Line length) dropdown set to '40', a 'Prikaži GTIN' (Show GTIN) checkbox, and a 'Šablon' (Template) field with the value 'config\templates\agent-invoice.vm'. The 'A4' section has a 'Šablon' field with the value 'config\templates\agent-invoice.jasper'. The 'Logotip' section has three buttons: 'Uvezi iz datoteke' (Import from file), 'Uvezi kroz URL' (Import via URL), and 'Obriši logotip' (Delete logo). At the bottom right, there are 'OK' and 'Odustani' (Cancel) buttons.

- **Dužina reda** – The default value is 40 characters per line.
- **Prikaži GTIN** – If this option is activated, then the barcode of the article will be printed before the name of the article, if the GTIN is sent in the request/file.
- **Logotip** – If this option is activated, the selected logo will be printed in the document header before printing the fiscal invoice. The logo can be imported from a file or via a URL, and supported formats are PNG and JPEG. Logotype printing is only available on POS printers.

## 2.3 Configure invoice printing

The second section refers to the possibility of printing a signed invoice after the response received from the PFR.



You must first select the "Štampanje omogućeno" option. The user can then select the type of printer and set all the necessary parameters for printing, according to the properties and functions supported by the printer.

Currently supported printer types:

- **ESC/POS Standard Printer** - It can be any thermal printer installed on the operating system and connected via a USB port. This is the safest option, as print management is left to the installed driver and operating system.
- **ESC/POS Serial Printer** - For thermal printers where the serial (com) port is the only option for connecting to a computer.

- **ESC/POS USB Printer** - For all thermal printers connected via USB port. This option should be used only if we do not have a driver available to install the printer on the operating system, in all other situations use the ESC / POS Standard Printer option.

After selecting the printer, you need to select the printing method. Two possibilities are offered here:

- **Prilagođeno** - universal printing method and every thermal printer should be able to print an invoice this way. It is only necessary to adjust the font size to the characteristics of the printer itself.
- **Nativno** - the invoice is printed by sending a series of ESC / POS commands. If this printing method is selected, the user must configure certain printer parameters. These parameters are not the same for every printer and it is necessary to consult the programming instructions supplied with the device. This way of printing has a slight advantage in quality compared to the "Prilagođeno" option, however, there are limitations because a large number of models do not support the entire existing set of ESC / POS instructions (eg some models do not support native code page for Serbian Cyrillic)

Options Pulse, T1 and T2 are related to sending signals to the port through which the printer is connected to the so-called cashbox, ie drawer / drawer for money (provided that the printer has this option at all). These parameters need to be adjusted according to the characteristics of the connected device.

### 3. Data exchange via file system

The exchange via the file system takes place according to the same principle as when it comes to the exchange via the web service, except that in this case, instead of HTTP transport, the file system is used.

Messages to be sent can be in XML or JSON format. The PFR Agent's response will be in the same format in which the message was forwarded.

A folder called "exchange" is used for messaging, which exists as a subfolder at the location where myLPFR is installed (eg C: \myLPFR\exchange). There are two subfolders within the "exchange" folder: "to-sdc" and "from-sdc". The "to-sdc" folder is used to forward commands to the PFR, while the "from-sdc" folder will find the answer after the command is processed. PFR Agent scans the "to-sdc" folder and each time a new file with the specified name appears in this folder, it will be loaded and its contents forwarded to the appropriate web service, and then the result obtained from the service will be forwarded back to the file system, in the "from-sdc" folder. According to the predefined protocol, LPFR always returns data in JSON format, however, this middleware also allows users to communicate through XML files, due to possible limitations of the technology in which ESIR was developed.



Commands are sent in files that have predefined names, and the name of the command corresponds to the name used in the technical documentation of the PU. The file name consists of the command name and the request identifier, in the format {CommandName} - {Identifier}. {Json / xml}. The format is binding, if the command does not respect this convention it will be rejected and an error will be received in the "from-sdc" folder.

The request identifier can be any unique alphanumeric and it is important that it exists due to the possibility of competitive operation of multiple ESIR instances. Each such instance should use a unique identifier, so that there is no "overwriting" of files in the same folder.

There are currently six types of commands:

- **Attention** - An empty file is sent in the format Attention- {identifier}. {Json / xml}. The command is used exclusively to check whether the LPFR is currently running, ie. whether it can receive requests. If so, you will also get an empty file called AttentionResponse- {identifier}. {Json / xml} in the "from-sdc" folder.
- **VerifyPin** - The file VerifyPin- {identifier}. {Json.xml} containing the PIN code is sent. The file VerifyPinResponse- {identifier} is returned. {Json / xml} which contains the status at execution (same as when calling via HTTP)
- **CreateInvoice** - The file CreateInvoice- {identifier} is sent. {Json / xml} with the contents of the account (same as for calls via HTTP). LPFR returns the signed invoice which will be returned in the file InvoiceResponse- {identifier}. {Json / xml} or an error in the file Error- {identifier}. {Json / xml}. The middleware component can also be configured to "unpack" the log (text representation of the account) and QR code in the "from-sdc" folder, thus making it easier for ESIR to parse and load print data. In this case, instead of the file InvoiceResponse- {identifier}. {Json / xml}, two other files will be obtained: Journal- {identifier} .txt and QR- {identifier}. {Gif / bmp / jpg / png}. If the PFR Agent is pre-configured to restore the account image, only the Receipt- {identifier} .png file will be saved in the "from-sdc" folder.
- **GetLastSignedInvoice** - an empty GetLastSignedInvoice file is sent. {json / xml}. The response is the last signed invoice with the forwarded identifier. All the rules that apply during the creation of the answer are the same as with the command **CreateInvoice**.
- **GetStatus** - An empty GetStatus file is being sent. {Json / xml}. The PFR response will be returned in the StatusResponse file {json / xml} with a description of the state the PFR is in (as in HTTP requests).
- **GetEnvironmentParameters** - An empty GetEnvironmentParameters file is being sent. {Json / xml}. The PFR response will be obtained in the EnvironmentParametersResponse file {json / xml} with the environment parameters (as for HTTP requests).

- **GetSubject** - Reading business data from Smart Card. Command is not part of the standard protocol. An empty GetSubject-`{identifier}.json/xml` file is being sent. The PFR response will be obtained in the file SubjectResponse-`{identifier}.json/xml` which contains basic information about the company, such as PIB, name, address, etc.

The PFR response is always recorded in the "from-sdc" folder and all text files are UTF-8 encoded. If there is any error in processing the request, the error data will be saved in the file Error-`{identifier}.json/xml`.

For a detailed description of the data forwarded and received in response from the PFR, as well as for error codes in request processing, consult the technical guide published by the PU, chapter POS to SDC Protocol:

<https://tap.sandbox.suf.purs.gov.rs/Help/view/1985194226/POS-to-SDC-Protocol/en-US>

POS to SDC Protocol codes								
Invoice type			Transaction type			Payment type		
0	Promet	Normal	0	Prodaja	Sale	0	Ostalo	Other
1	Predračun	Proforma	1	Refundacija	Refund	1	Gotovina	Cash
2	Kopija	Copy				2	Kartica	Card
3	Obuka	Training				3	Ček	Check
4	Avans	Advance				4	Virman	Wire Transfer
						5	Vaučer	Voucher
						6	IPS	Mobile Money

In addition to the error codes listed in the technical guide, the PFR Agent introduces five other possible codes, which can be obtained if there is a problem in communication via the file system:

- **3010** - Invalid filename - if the filename does not conform to the previously described convention
- **3020** - Unable to read file - if the file for some reason cannot be loaded (file system error, or if some other process is currently accessing the file)
- **3030** - Non-existent command - if the command name is not one of the above
- **3040** - Formatting error - if the file is not well formatted according to JSON / XML rules
- **3100** - PFR not available - if it is not possible to access PFR services via a configured address
- **3200** - Reports not activated - if the user has not activated the report module and a command to print reports has been sent

## 4. Examples of commands in XML format

### 4.1 Attention

Request : Attention-kasir1.xml (*empty file*)

Response : AttentionResponse-kasir1.xml (*empty file*)

### 4.2 VerifyPin

Request: VerifyPin-kasir1.xml

The content of the file is the PIN code of the card, e.g. 1234

Response: VerifyPinResponse-kasir1.xml

The content of the file is the status code (see the subsection Status and Error Codes in the technical guide for all status codes that can be returned in response to the LPFR)

Example:

```
<status>0100</status>
```

(PIN code is valid)

### 4.3 CreateInvoice

Request: CreateInvoice-kasir1.xml

Example:

```
<request>
  <cashier>kasir1</cashier>
  <invoiceNumber>POS2017/998</invoiceNumber>
  <dateAndTimeOfIssue>2021-10-29T23:20:23</dateAndTimeOfIssue>
  <invoiceType>0</invoiceType>
  <transactionType>0</transactionType>
  <buyerId>RS34564565</buyerId>
  <buyerCostCenterId>567546</buyerCostCenterId>
  <items>
    <item>
      <name>Test artikal 1</name>
      <quantity>1</quantity>
      <unitPrice>100.00</unitPrice>
      <totalAmount>100.00</totalAmount>
    </item>
  </items>
</request>
```

```

        <labels>
            <label>A</label>
        </labels>
    </item>
    <item>
        <name>Test artikal 2</name>
        <quantity>2</quantity>
        <unitPrice>150.00</unitPrice>
        <totalAmount>300.00</totalAmount>
        <labels>
            <label>A</label>
        </labels>
    </item>
    <item>
        <name>Test artikal 3</name>
        <quantity>1</quantity>
        <unitPrice>200.00</unitPrice>
        <totalAmount>200.00</totalAmount>
        <labels>
            <label>B</label>
        </labels>
    </item>
</items>
<payments>
    <payment>
        <paymentType>1</paymentType>
        <amount>600.00</amount>
    </payment>
</payments>
</request>

```

Response: InvoiceResponse-kasir1.xml

Example:

```

<response>
    <requestedBy>AHRQ2BQA</requestedBy>
    <signedBy>AHRQ2BQA</signedBy>
    <sdcdatetime>2021-10-29T23:20:10.614</sdcdatetime>
    <invoiceCounter>486/574ПП</invoiceCounter>
    <invoiceCounterExtension>ПП</invoiceCounterExtension>
    <invoiceNumber>AHRQ2BQA-AHRQ2BQA-574</invoiceNumber>

    <verificationUrl>https://sandbox.suf.purs.gov.rs/v/?v1=A0FIU1EyQ1FBQUhSUTJCUUE%2BAGAA5gEAAICNwWAAAAAAAAAB
    fWkp23YAAAPSUzM0NTY0NTY1HRX11B1sBGxwd1AD1CAMWdy%2BkULSFfk07YGBuWANHaP0r6knIbkWYuFVSA0e4...</verificationU
    rl>

    <verificationQRCode>R01G0D1hIwEjAFAAAAAAAP//ywAAAAAIwEjAUAC/4SPqcF97ZSBtMpJH7vI81RFnyeJnEguGxamlwm2H6sFq
    j3ncLcC5P7LxCA1V090zLGAQp9sWEywei2ckwilLZ/Xww1p/Wqn4W3Vaixfz9S20opdN+HpHVzcZcrZUrQ0qvTCjdU58YXyPOWwOiGxx
    cUkojYKK1HmbfHaBbpxdmUBhY6Zog2KHhGSDaayRppR3oI+kdrCumpqNqJCbnpkvNZ4YLm4v6W0rsODwDeMn1p1b...</verificatio
    nQRCode>
    <journal>===== ФИСКАЛНИ РАЧУН =====&#xd;
        123456789 &#xd;
        NAZIV FIRME &#xd;
        NAZIV PRODAJNOG MESTA &#xd;
        ADRESA PRODAJNOG MESTA &#xd;
        OPŠTINA &#xd;
        -----&#xd;
        ПИБ купца RS34564565&#xd;
        Опционо поље купца 567546&#xd;
        Касир kasir1&#xd;
        ЕСИР број POS2017/998&#xd;
        ЕСИР време 30.09.2021. 23:20:23&#xd;
        ----- ПРОМЕТ - ПРОДАЈА -----&#xd;

```

```

=====
      Артикли                                     &#xd;
=====&#xd;
Назив  Цена      Кол.      Укупно&#xd;
Test artikal 1 (A)                                     &#xd;
      100,00      1      100,00&#xd;
Test artikal 2 (A)                                     &#xd;
      150,00      2      300,00&#xd;
Test artikal 3 (B)                                     &#xd;
      200,00      1      200,00&#xd;
-----&#xd;
Укупан износ рачуна                                     600,00&#xd;
Готовина                                     600,00&#xd;
=====&#xd;
Ознака  Име      Стопа      Порез&#xd;
A      VAT      9.00%      33,03&#xd;
B      VAT      0.00%      0,00&#xd;
-----&#xd;
Укупан износ пореза                                     33,03&#xd;
=====&#xd;
ПФР време      29.10.2021. 23:20:10&#xd;
ПФР број рачуна  AHRQ2BQA-AHRQ2BQA-574&#xd;
Бројач рачуна    486/574ПП&#xd;
=====&#xd;
===== КРАЈ ФИСКАЛНОГ РАЧУНА =====&#xd;
</journal>
<totalCounter>574</totalCounter>
<transactionTypeCounter>486</transactionTypeCounter>
<totalAmount>600.0000</totalAmount>

<encryptedInternalData>HRX11B1sBGxWd1AD1CAMWdy+kULSfFk07YGBuWANHaP0r6knIbkWYuFVSA0e4+oVCoy3o5oeFvut0Hp6QL
A8wFGJLnmi141FXoncIk2Nt00mR2mC6tAAEQwkqYtnqrHZe5NjfpH886gXyjoS6kUo8S+bFz1SNINjqGgRuIR0XpqtXdHATX1zVq3Bk1M
GwLWI1tmIDEWD6PRq+ERI6yPwHr4BI1mFY0u3dpUPacSc1L9trPH3+3YNV/moKmW2fCX1QCbsNDnTA1PHM6jVsQj5fsjNPa4dZtyARFyZ
2IP4P90gWoUAjAq3l8B3fHH55sM47ctw1DegzSo0MFxk83kAlw==</encryptedInternalData>

<signature>TZxgj2rsEqNrGFwACsZ9tpGPPxeZS6iqBSx0oAWTxNwGH4WzEkX0/3RhbaJC7W5Z3bVwL2K3eBCPz1Yxna8PXYwcNYkVt0
wCr+KsFQVRkSBRaPQNj5acmqqt7q/86ahMV1Dj5Y/04ba9gyTSuRu+yL6mfYPeN8s3PFX3fLGhg9QCU1r8XJ80gZCZrRu3exgoxuIYE7n
rWt2RbDNHdCEEJa+cGSntMF7YzzFwiLu6K9deS10zssam4yxjrGVKdk8Yz4PnyuR9atHT82EZnk/606EKs5dcW746Qu4uh0ddzXmr31+Fm
VIjv/xVpWvAM37huxVwI3AbmnUwazymN9zRUTA==</signature>
<taxGroupRevision>2</taxGroupRevision>
<businessName>NAZIV FIRME</businessName>
<tin>123456789</tin>
<locationName>NAZIV PRODAJNOG MESTA</locationName>
<address>ADRESA PRODAJNOG MESTA</address>
<district>OPŠTINA</district>
<mrc>50-0001-A08CFDD863A5</mrc>
<messages>Success</messages>
<taxItems>
  <taxItem>
    <categoryType>0</categoryType>
    <label>A</label>
    <amount>33.0275</amount>
    <rate>9.00</rate>
    <categoryName>VAT</categoryName>
  </taxItem>
  <taxItem>
    <categoryType>0</categoryType>
    <label>B</label>
    <amount>0.0000</amount>
    <rate>0.00</rate>
    <categoryName>VAT</categoryName>
  </taxItem>
</taxItems>
</response>

```

If the "unpacking" of the log and QR code is pre-configured, the response will contain two other files instead of the previous XML file: Journal-kasir1.txt and QR-kasir1.gif. In this case, ESIR only needs to load the contents of the Journal-kasir1.txt file and print it on a printer together with the QR code obtained as an image in another response file.

## 4.4 GetLastSignedInvoice

Request: GetLastSignedInvoice-kasir1.xml (*empty file*)

Response: LastSignedInvoiceResponse-kasir1.xml

See command **Create Invoice** for an example of an answer

## 4.5 GetStatus

Request: GetStatus-kasir1.xml (*empty file*)

Response: StatusResponse-kasir1.xml

Response example:

```
<response>
  <isPinRequired>false</isPinRequired>
  <auditRequired>false</auditRequired>
  <sdcdatetime>2021-10-29T13:15:05.93</sdcdatetime>
  <lastInvoiceNumber>AHRQ2BQA-AHRQ2BQA-574</lastInvoiceNumber>
  <protocolVersion>1.0.0.0</protocolVersion>
  <secureElementVersion>1.1.0</secureElementVersion>
  <hardwareVersion></hardwareVersion>
  <softwareVersion>1.0.0</softwareVersion>
  <deviceSerialNumber>50-0001-A08CFDD863A5</deviceSerialNumber>
  <make>MyOffice D00</make>
  <model>myLPFR</model>
  <mssc/>
  <gsc>
    <gsc>0210</gsc>
  </gsc>
  <supportedLanguages>
    <supportedLanguages>sr-Cyrl-RS</supportedLanguages>
  </supportedLanguages>
  <uid>AHRQ2BQA</uid>
  <taxCoreApi>https://api.sandbox.suf.purs.gov.rs</taxCoreApi>
  <currentTaxRates>
    <validFrom>2021-07-22T09:55:24</validFrom>
    <groupId>2</groupId>
    <taxCategories>
      <taxCategory>
        <categoryId>8</categoryId>
        <name>ECAL</name>
        <categoryType>0</categoryType>
        <orderId>1</orderId>
        <taxRates>
          <taxRate>
            <rateId>9</rateId>
            <rate>11.00</rate>
            <label>F</label>
          </taxRate>
        </taxRates>
      </taxCategory>
    </taxCategories>
  </currentTaxRates>
```

```
<categoryId>9</categoryId>
<name>N-TAX</name>
<categoryType>0</categoryType>
<orderId>2</orderId>
<taxRates>
  <taxRate>
    <rateId>10</rateId>
    <rate>0.00</rate>
    <label>N</label>
  </taxRate>
</taxRates>
</taxCategory>
<taxCategory>
  <categoryId>10</categoryId>
  <name>PBL</name>
  <categoryType>2</categoryType>
  <orderId>3</orderId>
  <taxRates>
    <taxRate>
      <rateId>11</rateId>
      <rate>0.50</rate>
      <label>P</label>
    </taxRate>
  </taxRates>
</taxCategory>
<taxCategory>
  <categoryId>11</categoryId>
  <name>STT</name>
  <categoryType>0</categoryType>
  <orderId>4</orderId>
  <taxRates>
    <taxRate>
      <rateId>12</rateId>
      <rate>6.00</rate>
      <label>E</label>
    </taxRate>
  </taxRates>
</taxCategory>
<taxCategory>
  <categoryId>12</categoryId>
  <name>TOTL</name>
  <categoryType>1</categoryType>
  <orderId>5</orderId>
  <taxRates>
    <taxRate>
      <rateId>13</rateId>
      <rate>2.00</rate>
      <label>T</label>
    </taxRate>
  </taxRates>
</taxCategory>
<taxCategory>
  <categoryId>13</categoryId>
  <name>VAT</name>
  <categoryType>0</categoryType>
  <orderId>6</orderId>
  <taxRates>
    <taxRate>
      <rateId>14</rateId>
      <rate>9.00</rate>
      <label>A</label>
    </taxRate>
    <taxRate>
      <rateId>15</rateId>
      <rate>0.00</rate>
      <label>B</label>
    </taxRate>
  </taxRates>
</taxCategory>
```



```
</taxRates>
</taxCategory>
<taxCategory>
  <categoryId>14</categoryId>
  <name>VAT-EXCL</name>
  <categoryType>0</categoryType>
  <orderId>7</orderId>
  <taxRates>
    <taxRate>
      <rateId>16</rateId>
      <rate>0.00</rate>
      <label>C</label>
    </taxRate>
  </taxRates>
</taxCategory>
</taxCategories>
</currentTaxRates>
<allTaxRates>
  <taxGroup>
    <validFrom>2021-07-22T09:55:24</validFrom>
    <groupId>2</groupId>
    <taxCategories>
      <taxCategory>
        <categoryId>8</categoryId>
        <name>ECAL</name>
        <categoryType>0</categoryType>
        <orderId>1</orderId>
        <taxRates>
          <taxRate>
            <rateId>9</rateId>
            <rate>11.00</rate>
            <label>F</label>
          </taxRate>
        </taxRates>
      </taxCategory>
      <taxCategory>
        <categoryId>9</categoryId>
        <name>N-TAX</name>
        <categoryType>0</categoryType>
        <orderId>2</orderId>
        <taxRates>
          <taxRate>
            <rateId>10</rateId>
            <rate>0.00</rate>
            <label>N</label>
          </taxRate>
        </taxRates>
      </taxCategory>
      <taxCategory>
        <categoryId>10</categoryId>
        <name>PBL</name>
        <categoryType>2</categoryType>
        <orderId>3</orderId>
        <taxRates>
          <taxRate>
            <rateId>11</rateId>
            <rate>0.50</rate>
            <label>P</label>
          </taxRate>
        </taxRates>
      </taxCategory>
      <taxCategory>
        <categoryId>11</categoryId>
        <name>STT</name>
        <categoryType>0</categoryType>
        <orderId>4</orderId>
        <taxRates>
```

```
<taxRate>
  <rateId>12</rateId>
  <rate>6.00</rate>
  <label>E</label>
</taxRate>
</taxRates>
</taxCategory>
<taxCategory>
  <categoryId>12</categoryId>
  <name>TOTL</name>
  <categoryType>1</categoryType>
  <orderId>5</orderId>
  <taxRates>
    <taxRate>
      <rateId>13</rateId>
      <rate>2.00</rate>
      <label>T</label>
    </taxRate>
  </taxRates>
</taxCategory>
<taxCategory>
  <categoryId>13</categoryId>
  <name>VAT</name>
  <categoryType>0</categoryType>
  <orderId>6</orderId>
  <taxRates>
    <taxRate>
      <rateId>14</rateId>
      <rate>9.00</rate>
      <label>A</label>
    </taxRate>
    <taxRate>
      <rateId>15</rateId>
      <rate>0.00</rate>
      <label>B</label>
    </taxRate>
  </taxRates>
</taxCategory>
<taxCategory>
  <categoryId>14</categoryId>
  <name>VAT-EXCL</name>
  <categoryType>0</categoryType>
  <orderId>7</orderId>
  <taxRates>
    <taxRate>
      <rateId>16</rateId>
      <rate>0.00</rate>
      <label>C</label>
    </taxRate>
  </taxRates>
</taxCategory>
</taxCategories>
</taxGroup>
<taxGroup>
  <validFrom>2021-07-15T23:08:09</validFrom>
  <groupId>1</groupId>
  <taxCategories>
    <taxCategory>
      <categoryId>1</categoryId>
      <name>ECAL</name>
      <categoryType>0</categoryType>
      <orderId>1</orderId>
      <taxRates>
        <taxRate>
          <rateId>1</rateId>
          <rate>10.00</rate>
          <label>F</label>
        </taxRate>
      </taxRates>
    </taxCategory>
  </taxCategories>
</taxGroup>
```

```
        </taxRate>
    </taxRates>
</taxCategory>
<taxCategory>
    <categoryId>2</categoryId>
    <name>N-TAX</name>
    <categoryType>0</categoryType>
    <orderId>2</orderId>
    <taxRates>
        <taxRate>
            <rateId>2</rateId>
            <rate>0.00</rate>
            <label>N</label>
        </taxRate>
    </taxRates>
</taxCategory>
<taxCategory>
    <categoryId>3</categoryId>
    <name>PBL</name>
    <categoryType>2</categoryType>
    <orderId>3</orderId>
    <taxRates>
        <taxRate>
            <rateId>3</rateId>
            <rate>0.50</rate>
            <label>P</label>
        </taxRate>
    </taxRates>
</taxCategory>
<taxCategory>
    <categoryId>4</categoryId>
    <name>STT</name>
    <categoryType>0</categoryType>
    <orderId>4</orderId>
    <taxRates>
        <taxRate>
            <rateId>4</rateId>
            <rate>6.00</rate>
            <label>E</label>
        </taxRate>
    </taxRates>
</taxCategory>
<taxCategory>
    <categoryId>5</categoryId>
    <name>TOTL</name>
    <categoryType>1</categoryType>
    <orderId>5</orderId>
    <taxRates>
        <taxRate>
            <rateId>5</rateId>
            <rate>2.00</rate>
            <label>T</label>
        </taxRate>
    </taxRates>
</taxCategory>
<taxCategory>
    <categoryId>6</categoryId>
    <name>VAT</name>
    <categoryType>0</categoryType>
    <orderId>6</orderId>
    <taxRates>
        <taxRate>
            <rateId>6</rateId>
            <rate>9.00</rate>
            <label>A</label>
        </taxRate>
    </taxRates>
</taxCategory>
```

```
        <rateId>7</rateId>
        <rate>0.00</rate>
        <label>B</label>
      </taxRate>
    </taxRates>
  </taxCategory>
<taxCategory>
  <categoryId>7</categoryId>
  <name>VAT-EXCL</name>
  <categoryType>0</categoryType>
  <orderId>7</orderId>
  <taxRates>
    <taxRate>
      <rateId>8</rateId>
      <rate>0.00</rate>
      <label>C</label>
    </taxRate>
  </taxRates>
</taxCategory>
</taxCategories>
</taxGroup>
</allTaxRates>
</response>
```

## 4.6 GetEnvironmentParameters

Request : GetEnvironmentParameters-kasir1.xml

Response : EnvironmentParametersResponse-kasir1.xml

Response example :

```
<response>
  <organizationName>Министарство финансија - Пореска управа - Централa</organizationName>
  <serverTimeZone>Central Europe Standard Time</serverTimeZone>
  <street>Саве Машковића 3-5</street>
  <city>Београд</city>
  <country>RS</country>
  <environmentName>СУФ Развој</environmentName>
  <logo>https://sandbox.suf.purs.gov.rs:443/DownloadContent/TALogo.png</logo>
  <ntpServer>http://0.pool.ntp.org:80/</ntpServer>
  <endpoints>
    <taxpayerAdminPortal>https://tap.sandbox.suf.purs.gov.rs:443/</taxpayerAdminPortal>
    <taxCoreApi>https://api.sandbox.suf.purs.gov.rs:443/</taxCoreApi>
    <vsdc>https://vsdc.sandbox.suf.purs.gov.rs:443/</vsdc>
    <root>https://sandbox.suf.purs.gov.rs:443/v/?v1=</root>
  </endpoints>
  <supportedLanguages>
    <supportedLanguages>sr-Cyrl-RS</supportedLanguages>
    <supportedLanguages>en-US</supportedLanguages>
  </supportedLanguages>
</response>
```

## 4.7 GetSubject

Request : GetSubject-kasir1.xml

Response : SubjectResponse-kasir1.xml

Response example :

```
<response>
  <uid>AHRQ2BQA</uid>
  <tin>123456789</tin>
  <businessName>NAZIV FIRME</businessName>
  <locationName>NAZIV PRODAJNOG MESTA</locationName>
  <address>ADRESA PRODAJNOG MESTA</address>
  <city>GRAD</city>
  <district>OPŠTINA</district>
</response>
```

## 5. Intermediary web services

*PFR Agent* introduces two intermediary web services to enable ESIR solutions that can use HTTP transport to easily print a signed invoice.

### 5.1 Creating an invoice

The service is located at the endpoint `/agent/v3/invoices` and it is accessed through POST methods (e.g. <http://localhost:8989/agent/v3/invoices>). The protocol according to which the communication takes place is identical to the standard service `/api/v3/invoices`. The PFR Agent mediates in the exchange between the two ends and in no way changes the data received from the ESIR, but only forwards them in the received form to the LPFR. After receiving the answer, the result is returned to ESIR, and the journal is sent to print, if the print parameters are pre-configured.

### 5.2 Getting the last signed invoice

The service is located at the endpoint `/agent/v3/invoices/{ request_identifier }` and is accessed through the GET method (e.g. <http://localhost:8989/agent/v3/invoices/kasir1>). The protocol is identical to the standard `/api/v3/invoices` service. The PFR Agent delivers the last signed invoice according to the request identifier, returns it to ESIR, and then sends the log to print, if the print parameters are pre-configured.

### 5.3 Providing an image of a previously signed invoice

The service is located at the endpoint `/agent/v3/receipts/{request_identifier}` and is accessed via the GET method (e.g. [http://localhost:8989/agent/v3/receipts/{request\\_identifier}](http://localhost:8989/agent/v3/receipts/{request_identifier})). This service can be useful if you want to save an image of a previously issued invoice or if you want to print an invoice from ESIR (POS). The image that method returns is in PNG format.

## 6. Model extension

When sending a request for invoice fiscalization, it is possible to pass several additional parameters that will be taken into account when generating / printing the journal. These are the following parameters:

- **print** - whether you want to print the invoice, or just sign. If the parameter value is false, the PFR Agent will skip sending the log to print. This parameter can be useful in situations where ESIR wants to take control of the printing of certain types of accounts.
- **message** - adds a text message to the customer after the last line of the journal.
- **advance** - adds information about the advance payment to the journal. The parameter should be used only in the case of the final invoice, in order to show the closing of the account with a previously paid advance.

- **advanceTax** - adds information on advance payment tax. The parameter should only be used in combination with the advance parameter.

All parameters are optional and are added within a special request element, called **journalOptions**, which is also optional.

#### XML example account with additional parameters:

```
<request>
  <cashier>kasir1</cashier>
  <invoiceNumber>POS2017/998</invoiceNumber>
  <dateAndTimeOfIssue>2021-10-29T23:20:23</dateAndTimeOfIssue>
  <invoiceType>0</invoiceType>
  <transactionType>0</transactionType>
  <items>
    <item>
      <name>Test artikal</name>
      <quantity>1</quantity>
      <unitPrice>1000.00</unitPrice>
      <totalAmount>1000.00</totalAmount>
      <labels>
        <label>A</label>
      </labels>
    </item>
  </items>
  <payments>
    <payment>
      <paymentType>1</paymentType>
      <amount>500.00</amount>
    </payment>
  </payments>
  <journalOptions>
    <print>false</print>
    <message>PORUKA ZA KUPCA</message>
    <advance>500.00</advance>
    <advanceTax>100.00</advanceTax>
  </journalOptions>
</request>
```

#### JSON example account with additional parameters:

```
{
  "cashier": "kasir1",
  "invoiceNumber": "POS2017/998",
  "dateAndTimeOfIssue": "2021-10-29T23:20:23",
  "invoiceType": "0",
  "transactionType": "0",
  "items": [
    {
      "name": "Test artikal",
      "quantity": "1",
      "unitPrice": "1000.00",
      "totalAmount": "1000.00",
      "labels": ["A"]
    }
  ],
  "payment": [
    {
      "paymentType": "1",
```

```

    "amount": "500.00"
  },
  "journalOptions": {
    "print": "false",
    "message": "PORUKA ZA KUPCA",
    "advance": "500.00",
    "advanceTax": "100.00"
  }
}

```

## 7. Encoding and tax rates

We have noticed that the labels for tax rates in the production environment differ significantly from the labels in the development environment to which ESIR developers have access when testing and certifying their solutions. It is necessary to pay attention to the fact that in the production environment, tax rates are marked in Cyrillic letters, in contrast to the development environment where Latin is used, so the following labels are currently in use:

**A** – Not in VAT

**Г** – Without VAT

**Ђ** – General rate (20%)

**Е** – Special rate (10%)

If the ESIR developer does not have the ability to properly set the encoding for Cyrillic, the rate mark of the item on the invoice can be forwarded in unicode notation. The table below contains the letters of the Serbian Cyrillic alphabet with the corresponding unicode values.

А	Б	В	Г	Д	Ђ
0410	0411	0412	0413	0414	0402
Е	Ж	З	И	Ј	К
0415	0416	0417	0418	0408	041A
Л	Љ	М	Н	Њ	О
041B	0409	041C	041D	040A	041E
П	Р	С	Т	Ћ	У
041F	0420	0421	0422	040B	0423
Ф	Х	Ц	Ч	Џ	Ш
0424	0425	0426	0427	040F	0428



The table shows that e.g. letter Ъ has a numeric value of 0402. When sending a tag, it is necessary to escape the value, so that during deserialization it would be clear that it is a number that represents a unicode character. In the case of JSON format, the value is escaped by adding the sequence "\ u" in front, and "& # x" in front of and ";" behind in the case of XML format.

Oznaka	JSON	XML
A	\u0410	&#x0410;
Г	\u0413	&#x0413;
Ђ	\u0402	&#x0402;
Е	\u0415	&#x0415;

Example JSON request with rate label in unicode format:

```
{
  "payment" : [ {
    "paymentType" : 1,
    "amount" : 100.00
  } ],
  "cashier" : "123456789",
  "invoiceNumber" : "POS2017/998",
  "invoiceType" : 0,
  "buyerId" : "RS34564565",
  "items" : [ {
    "name" : "Test artikla",
    "labels" : [ "\u0402" ],
    "totalAmount" : 100.00,
    "quantity" : 1,
    "unitPrice" : 100.00
  } ],
  "transactionType" : 0,
  "buyerCostCenterId" : "567546",
  "dateAndTimeOfIssue" : "2021-09-30T23:20:23"
}
```

Example XML request with rate label in unicode format:

```
<request>
  <cashier>123456789</cashier>
  <invoiceNumber>POS2017/998</invoiceNumber>
  <dateAndTimeOfIssue>2021-09-30T23:20:23</dateAndTimeOfIssue>
  <invoiceType>0</invoiceType>
```

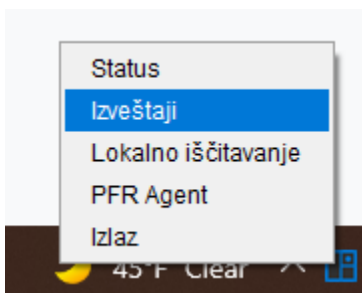
```
<transactionType>0</transactionType>
<buyerId>RS34564565</buyerId>
<buyerCostCenterId>567546</buyerCostCenterId>
<items>
  <item>
    <name>Test artikal 1</name>
    <quantity>1</quantity>
    <unitPrice>100.00</unitPrice>
    <totalAmount>100.00</totalAmount>
    <labels>
      <label>&#x0402;</label>
    </labels>
  </item>
</items>
<payments>
  <payment>
    <paymentType>1</paymentType>
    <amount>1000</amount>
  </payment>
</payments>
</request>
```

## 8. Reports

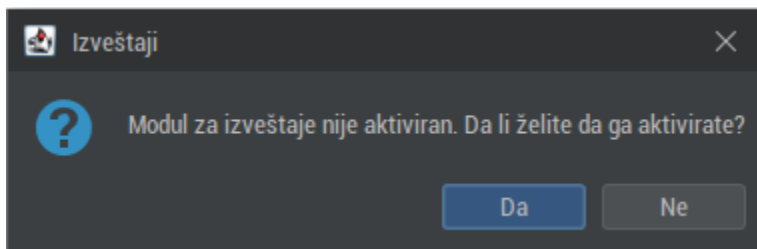
**NOTE:**The reports refer exclusively to the account signed by the observed myLPFR instance

The reports are part of the functionality that myLPFR adds to existing functions and are not defined in the Technical Guide for LPFR component implementation. Three types of reports were supported, following the example of the previous model implemented by fiscal cash registers: Cross-section of the situation, Daily Report and Periodic Report.

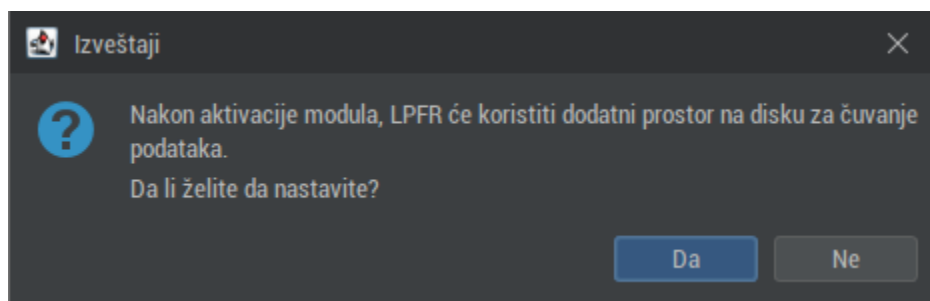
This module needs to be activated after installation to make print commands available. Activation is done by selecting the Reports option from the main menu of the application.



After selecting this option, it is necessary to answer in the affirmative to the question shown in the dialogue that follows.



The user is then informed that myLPFR will need to use additional disk space for data storage. The user must also answer in the affirmative to the question of whether he wishes to allow such a possibility.



From the moment the report is activated, myLPFR will record all generated Daily Reports in the local database and will use them when calculating the amount on the *Periodic Report*.

## 8.1 Web services for generating and printing reports

**Cross-section of the situation and Daily Report** - the service is located at the endpoint `/agent/v3/reports/daily` and is accessed through the GET method (ie. <http://localhost:8989/agent/v3/reports/daily>). The *CloseDay* header parameter, whose value is of type *boolean*, can also be passed to this service. If the value is *false*, feedback report will be *Cross-section of the situation*, while if the value is *true* report will be type *Daily report*. After the report is generated, the result is returned to the ESIR, and the journal is sent to print, if the print parameters have been pre-configured.

**Periodic report** – service is located at the endpoint `/agent/v3/reports/periodic` and is accessed through the GET method (ie. <http://localhost:8989/agent/v3/reports/periodic>). Two header parameters can be passed to this service: *StartDate* i *EndDate*. *StartDate* is the start date of the calculation, and *EndDate* is the end date of the calculation. Both dates are forwarded in the format *yyyy-MM-dd* (ie. 2021-12-31). After the report is generated, the result is returned to the ESIR, and the journal is sent to print, if the print parameters have been pre-configured.

It is best to consult for more detailed instructions *Swagger* documentation, which is available after starting the application, at <http://localhost:8989/swagger-ui.html>.

## 8.2 File command system for generating and printing reports

To generate all three types of reports, a request is passed in a file with the format name `GetReport-{identifikator}.json/xml`. This file, in its structure, may contain the following elements:

- **print** – message to *PFR Agent* whether a report should be printed - can have value *true* or *false*, and default values is *true*. The report will be printed if the value is *true* and the printer is configured in the PFR Agent setup section
- **reportType** - report type - can have values *Daily* ili *Periodic*, ie. 0 or 1 respectively
- **closeDay** - closing day - can have value *true* or *false* and should exist only in the situation when *reportType=Daily*. This value is used to control whether the report type will be Cross-Section Balance or Daily Report. If the value is *true*, we get a Daily Report.
- **startDate** - start date for the period - value in format *yyyy-MM-dd* (ie. 2021-12-31) and should exist only in the situation when *reportType=Periodic*
- **endDate** - end date for period - value in format *yyyy-MM-dd* (npr. 2021-12-31) and should exist only in the situation when *reportType=Periodic*

If the reports are not previously activated, the PFR Agent will return an error with code 3200.

### Example of a *Cross-section request*

#### XML:

File name: `GetReport-kasir1.xml`

```
<request>
  <reportType>Daily</reportType>
  <closeDay>>false</closeDay>
</request>
```

#### JSON:

File name: `GetReport-kasir1.json`

```
{
  "reportType": "Daily",
  "closeDay": "false"
}
```

**Example of a request for a *Daily Report*****XML:**

File name: GetReport-kasir1.xml

```
<request>
  <reportType>Daily</reportType>
  <closeDay>true</closeDay>
</request>
```

**JSON:**

File name: GetReport-kasir1.json

```
{
  "reportType": "Daily",
  "closeDay": "true"
}
```

**Example of a request for a *Periodic Report*****XML:**

File name: GetReport-kasir1.xml

```
<request>
  <reportType>Periodic</reportType>
  <startDate>2022-01-01</startDate>
  <endDate>2022-03-31</endDate>
</request>
```

**JSON:**

File name: GetReport-kasir1.json

```
{  
  "reportType": "Periodic",  
  "startDate": "2022-01-01",  
  "endDate": "2022-03-01"  
}
```

The response format depends on how the option *Način eksportovanja računa* is configured in *PFR Agent* settings, so that the answer can be obtained in the original JSON/XML format (ReportResponse-{identifikator}.json/xml), in text format (ReportResponse-{identifikator}.txt and ReportJournal-{identifikator}.txt), or as a picture (Report-{identifikator}.png). The answer includes summary values, grouped by method of payment, as well as by tax rates. The structure of the answers is almost identical for all three types of reports. *Daily Report* differs from the other two types in that it also contains information on the ordinal number of the report.

### 8.3 Response structure (web services and command file system)

After generating (and printing) the report, the PFR Agent returns the response to the ESIR in JSON or XML format. The answer elements are as follows:

- **uid** - JID of card (string)
- **dateTime** - report generation time, in format *yyyy-MM-ddTHH:mm:ss*
- **number** - ordinal number of the report (int) – exists only in *Daily Report*
- **tin** - PIB (string)
- **businessName** - the name of the business entity (string)
- **locationName** - the name of the point of sale (string)
- **address** - point of sale address (string)
- **district** - municipality (string)
- **total** - element type *Summary* which contains the total amounts of payments and taxes for *sale* and *advance*
- **perTransactionType** - array of type elements *TransactionTypeSummary*, where each element contains the total amounts of payments and taxes by individual type of account/transaction
- **journal** - textual representation of the report

*Summary* - total amount

- **invoiceCount** - total number of invoices issued
- **payments** - array of type elements *Payment*, where each contains the amount of payments by individual method of payment

- **taxItems** - array of type elements *TaxItem*, where each contains the sum of the value of the tax at the individual rate

*TransactionTypeSummary* - amount by type of account / transaction

- **invoiceCount** - total number of invoices issued
- **invoiceType** – invoice type (int) - the value corresponds to the types of accounts defined in the Technical Guide and can be 0 or 4 (sale or advance)
- **transactionType** – transaction type (int) - the value corresponds to the types of transactions defined in the Technical Guide and can be 0 or 1 (sale or refund)
- **payments** – array of type elements *Payment*, where each contains the amount of payments by individual method of payment
- **taxItems** - array of type elements *TaxItem*, where each contains the summary value of the tax at the individual tax rate