**Fiscalization service**

(Version v4.2)

Technical specification

**Versions**

|  |  |
| --- | --- |
| **Version** | **Change description** |
| v1 | Draft version |
| v2 | Convergence with the Functional specification v2 |
| v3 | New type ADVANCE of payment method added.  ConsTax element removed from invoice fiscalization message. |
| v4 | 1. Sample for ENU maintainer code changed from [0-9] {3} to [az] {2} [0-9] {3} [aZ] {2} [0-9] {3} (Chapter 3.5.1.9). 2. Sample for Date and Time changed from [0-9] {4} - [0-9] {2} - [0-9] {2} T [0-9] {2}, [0-9] {2}, [0-9] {2} [+ -] [0-9] {2}, [0-9] {2} u [0-9] {4} - [0-9] {2 } - [0-9] {2} T [0-9] {2}: [0-9] {2}: [0-9] {2} ([+ -] [0-9] {2} : [0-9] {2} | Z) 3. Values in the Post Delivery Type fields expanded with BUSINESSNEED values. 4. A prefix in the form of a business premises code has been added to the sample for the Account Number (Chapter 3.7.1.10). 5. Voucher number sample changed from [1-9] [0-9] {0,7} - [0-9] {4} - [0-9] {10} to [1-9] [0-9 ] {0.7} - [0-9] {4} - [0-9] {8}. 6. The option to send up to 4 decimal places has been added to decimal places for invoice items. 7. Changed rebate pattern of item from 12 to ([1-9] [0-9] \* | 0) \. [0-9] {2,4} | 0) (Chapter 3.7.1.82). 8. Updated list of values for exemptions (Chapters 3.7.1.86 and 3.7.1.94). 9. XSD schema changed (Chapter 7). |
| v4.1 | 1. Field taxPeriod added (Chapter 3.7.1.51) 2. URL for QR code generation changed from https://efi.tax.gov.me/ic/#/verify to https://mapr.tax.gov.me/ic/#/verify (Chapter 3.7.7). 3. ENU registration added for non-cash invoices (Chapter 3.5). 4. Condition for registration and change of initial cash deposit added (Chapter 3.6). 5. Sending of ENU code added as mandatory for fiscalization of cash and non-cash invoices. 6. Invoice number format changed (Chapter3.7.1.10). 7. Field "Reverse charge" is always sent false for invoice fiscalization (Chapter3.7.1.29). 8. Types of payment methods changed (Chapter 3.7.1.41). 9. Typos for IKOF data elements corrected (Chapter 4.3.2.1). 10. XSD schema changed according to the above previous changes (Chapter 7). |
| v4.2 | 1. All examples that include decimal places fixed (dot is used everywhere instead of comma). 2. Field obligation in field descriptions for Amount without tax and Amount of margin corrected. 3. Emphasized and detailed description of the field Same taxes (Chapter 3.7.1.89). 4. Description of how to round amounts on items, and tolerance on group and total amounts added (Chapter3.7.1.66). |

**Additional documentation**

|  |  |
| --- | --- |
| **Document name** | **Description** |
| Fiscalization service – Functional specification | All processes are described |

CONTENT

[1. INTODUCTION 4](#_Toc58993770)

[1.1. ABBREVIATIONS 4](#_Toc58993771)

[1.2. TERMINOLOGY 4](#_Toc58993772)

[2. ENVIRONMENTS 6](#_Toc58993773)

[2.1. PREPARATION WORKS FOR SERVICE USAGE 6](#_Toc58993774)

[2.2. TOPOLOGY 6](#_Toc58993775)

[2.2.1. CIS access point 6](#_Toc58993776)

[2.2.2. Internet connection 6](#_Toc58993777)

[2.2.3. Client information system 6](#_Toc58993778)

[2.3. CONDITION FOR CONNECTION WITH CIS 7](#_Toc58993779)

[2.3.1. Network preconditions and recommendations 7](#_Toc58993780)

[2.3.2. Security preconditions 7](#_Toc58993781)

[2.3.3. Application preconditions 8](#_Toc58993782)

[3. INTERFACE 9](#_Toc58993783)

[3.1. INTERFACE VERSIONING 9](#_Toc58993784)

[3.2. DATA MESSAGE CODING 9](#_Toc58993785)

[3.3. DATA MESSAGE STRUCTURE 10](#_Toc58993786)

[3.4. CONTROLS 10](#_Toc58993787)

[3.5. ENU REGISTRATION 11](#_Toc58993788)

[3.5.1. REGISTER ENU REQUEST MESSAGE 12](#_Toc58993789)

[3.5.2. REGISTER ENU RESPONSE MESSAGE 15](#_Toc58993790)

[3.5.3. MANDATORY CONTROLS 16](#_Toc58993791)

[3.5.4. ERROR MESSAGE 17](#_Toc58993792)

[3.5.5. XML EXAMPLES 17](#_Toc58993793)

[3.6. CASH DEPOSIT REGISTRATION 18](#_Toc58993794)

[3.6.1. REGISTER CASH DEPOSIT REQUEST MESSAGE 18](#_Toc58993795)

[3.6.2. REGISTER CASH DEPOSIT RESPONSE MESSAGE 22](#_Toc58993796)

[3.6.3. MANDATORY CONTROLS 23](#_Toc58993797)

[3.6.4. ERROR MESSAGE 24](#_Toc58993798)

[3.6.5. XML EXAMPLE 24](#_Toc58993799)

[3.7. INVOICE REGISTRATION AND FISCALIZATION 25](#_Toc58993800)

[3.7.1. REGISTER INVOICE REQUEST MESSAGE 26](#_Toc58993801)

[3.7.2. REGISTER INVOICE RESPONSE MESSAGE 54](#_Toc58993802)

[3.7.3. MANDATORY CONTROLS 56](#_Toc58993803)

[3.7.4. ERROR MESSAGE 57](#_Toc58993804)

[3.7.5. XML EXAMPLE 57](#_Toc58993805)

[3.7.6. DATA EXPORT AND UPLOAD 58](#_Toc58993806)

[3.7.7. QR CODE GENERATION 61](#_Toc58993807)

[3.8. ERROR MESSAGE 63](#_Toc58993808)

[3.8.1. XML FORMAT 63](#_Toc58993809)

[3.8.2. ERROR CODES 65](#_Toc58993810)

[3.8.3. XML EXAMPLE 66](#_Toc58993811)

[4. SECURITY 67](#_Toc58993812)

[4.1. DIGITAL CERTIFICATES 67](#_Toc58993813)

[4.2. TRANSPORT SECURITY 67](#_Toc58993814)

[4.3. MESSAGE SECURITY 67](#_Toc58993815)

[4.3.1. ELECTRONIC REQUEST AND RESPONS MESSAGE SIGNING 67](#_Toc58993816)

[4.3.2. IKOF DATA ELEMENT 68](#_Toc58993817)

[5. ANEX 1 – CODE EXAMPLES 71](#_Toc58993818)

[5.1. IKOF GENERATION CODE 71](#_Toc58993819)

[5.1.1. JAVA EXAMPLE 71](#_Toc58993820)

[5.1.2. C# EXAMPLE 71](#_Toc58993821)

[5.2. SIGNATURE GENERATION CODE 72](#_Toc58993822)

[5.2.1. JAVA EXAMPLE 72](#_Toc58993823)

[5.2.2. C# EXAMPLE 73](#_Toc58993824)

[6. ANEX 2 – WSDL version 1 75](#_Toc58993825)

[7. ANEX 3 – XSD version 1 77](#_Toc58993826)

# INTODUCTION

This document provides a description of the data interface for invoice and acknowledgement of data messages containing information on sales which the taxpayers are obliged to send for every sale made and subject to registration of Sales, i.e. invoices issued.

Files containing definition of the XML schema and the Web service (WSDL), which describe the structure of the registered invoice data messages and the Web service used to receive them are provided as Annexes 2 and 3 to this document.

This document provides specification for the fiscalization service version 1.

Old, deprecated service version will be available for 4 weeks after the new version is released. Date when a new version is released can be found at Tax administration website.

## ABBREVIATIONS

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
| CA | Registered certificate authority |
| CIS | EFI Central information system; Central invoice register |
| CRL | List of recalled certificates |
| JIKR (engl. FIC) | Unique invoice code |
| GUID | Global unique identifier |
| IKOF (eng. IIC) | Taxpayer identifying code |
| JMB | National unique identifying number |
| PIB | Tax identifying number (TIN) |
| OCSP | Certificate status online protocol |
| SEP (engl. SCP) | Self-care EFI portal – web portal that the taxpayer uses for adding specific data, and for other uses in fiscalization process |
| SOAP | XML message exchange protocol as specified on <https://www.w3.org/TR/soap/> |
| ENU (engl. TCR) | Taxpayer's electronic device. |
| ENU code (engl. *TCRCode*) | Electronic cash device code |
| UC | Use case diagrams |
| UJI (engl. UUID) | Universal unique identifier |
| WSDL | XML based language that provides web service description, as specified on <http://www.w3.org/TR/wsdl> |
| XML schema | XML based language used for defining XML documents structure as specified on <http://www.w3.org/TR/xmlschema11-1/> and on <https://www.w3.org/TR/xmlschema11-2/> |

Table 1: Abbreviations

## TERMINOLOGY

|  |  |
| --- | --- |
| **Expression** | **Expression meaning** |
| Electronic response message | Data structure in defined format that was defined by the organization in charge of financial affairs, which contains fiscal identifying code (JIKR) and is used as invoice confirmation, and formal correctness of the recorded response message. |
| Error message | Data structure in defined format that was defined by the organization in charge of financial affairs, that contains the error code and the error description in human readable text, as like as the answer on the received electronic message on the invoice that contains important errors that are disabling its processing, or when new blocking error appears. |
| Invoice | Invoice is a proof of a sale that the taxpayer gave (in paper, or in electronic message) to a customer or a subject that is buying the goods, and contains all the necessary data of quantities and amounts of sale and sold items.  Invoice is each document that complies with the conditions proposed by the draft of the Law on invoices and system for traffic overview. |
| Invoice issuer | A person who is issuing the invoice. He is responsible for fiscalization of the invoices in the central invoice register. In most cases that person is also the seller of the goods and services, but in self-vending machine cases, the buyer is also the issuer. |
| Recorded invoice | The invoice that is recorded in the central invoice register and that contains the fiscal identifying code. |
| Electronic message on the recorded invoice | Data structure in defined format that was defined by the organization in charge of financial affairs that contains data on selling and other needed technical information. This is a whole XML message that contains information described in relevant web service standards: SOAP / WSDL. |
| EFI self-care portal (SEP) | SEP is an application for taxpayers that gives the support for the invoice fiscalization processes. |
| ENU / taxpayer's electronic device | ENU or electronic cash device is the taxpayer's machine that sends the data on recorded invoices to the Tax administration.  The taxpayer's ENU or electronic invoice device is a device on the side of the taxpayer that sends data on registered invoices to the tax authority. This can mean, depending on the context, end devices such as ENU, or additional software and hardware that actually send data on recorded invoices. Electronic messages contains an item labeled "Electronic Billing Device Code" to denote an end device (Electronic Billing Device). In other parts of the text, this term usually means an end device and the corresponding software and hardware that send electronic messages. |

Table 2: Terminology

# ENVIRONMENTS

The government will publish Web service addresses for two types of environments: production environment and one or more test environments:

* **Non-production environment** will be used by software developers (developing software for ENU), and by ENU end users. Sending a data message to the non-production environment shall not be considered sending of registered invoice information. The JIKR returned by the non-production environment is not a valid JIKR (it is different per prefix).
* **Production environment** is intended for the taxpayers and will be used for routine operations, i.e. receipt and acknowledgement of data messages containing information on registered sales.

Endpoints:

o Testing environment:

<https://efitest.tax.gov.me/fs-v1>

o Production environment:

<https://efi.tax.gov.me/fs-v1>

## PREPARATION WORKS FOR SERVICE USAGE

Details can be found in Fiscalization service – Functional specification document v2.

## TOPOLOGY

Users access the CIS by initiating 1-way TLS connection. Clients exchange messages with Tax administration’s access point using TLS channel by described procedure. Data exchange is synchronous, meaning access point answers on user’s request immediately. Request and response messages formats are specified through XML schema.

### CIS access point

Implementation and maintenance of the access point is a TBD’s task. TBD company will provide its users connection to the access point in two environments: production and test.

### Internet connection

Access point will be available through internet networks in HTTPS protocol.

### Client information system

Clients are obliged to provide hardware and software support for messages exchange with access point. As shown on image below, there is no mediatory component development planned. Development of the hardware-software solutions is in client’s domain of business. Client is also obliged to secure internet connection to CIS access point with needed bandwidth. Platform choice and software solution implementation is in client’s domain and such information is not needed to be reported to TBD company.

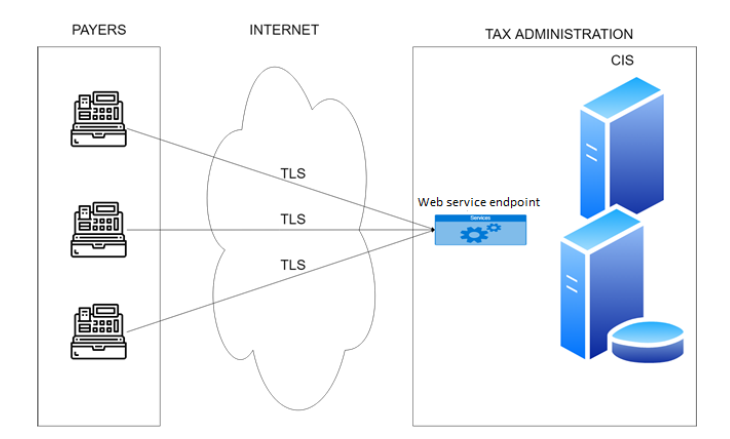


Image 1:Client information system

## CONDITION FOR CONNECTION WITH CIS

Central information system (CIS) of the Tax Administration will be available in two environments: production and test.

Connection conditions are similar but differ in addresses of their access points and certificates. Both, in production and test environment certificates are issued by Registered CA. Environments are not different in its functionality (besides new functionalities development), only difference is in data – test environment uses test data.

### Network preconditions and recommendations

To connect to the CIS of the Tax Administration, client system needs to fulfil these conditions:

|  |  |
| --- | --- |
| Network type | Internet |
| Recommended open TCP ports to CIS | 443 |

Table 3

Network recommendations for client system are:

|  |  |
| --- | --- |
| **Link characteristics** | **Permanent symmetric link** |
| Bandwidth | 2 Mb/s at minimum (up to 40 messages per second) |

Table 4

### Security preconditions

All communication with CIS of the Tax Administration is protected by 1-way TLS encryption at the transport layer. In production environment CIS presents itself to client with a TLS certificate issued by Registered CA.

|  |  |
| --- | --- |
| Protection at the transportation layer | HTTPS (at least TLS v1.1 and v1.2, AES\_256 encryption) |
| Certificates for the electronic signing | Certificate type: application digital certificate for the fiscalization |

Table 5

### Application preconditions

CIS functionality is available to its clients using web-service technology. For that reason, client´s application (or infrastructure, depending on realization) needs to fulfil these preconditions:

|  |  |
| --- | --- |
| Client creation standards | WS-1 |
| Service type | Document-literal |
| Application protocol | SOAP / HTTPS (SOAP 1.1) |
| Code site of the request message XML | UTF-8 |

Table 6

# INTERFACE

Interface for exchanging the data between the taxpayer and CIS regarding the fiscalization will be SOAP web service. Messages are in XML format according to the standards of SOAP messages.

The web service has several operations which will be used by the taxpayer who needs to do the fiscalization of the invoices. Invoices are issued by the electronic device represented by its code. The code is assigned in operation of registration of electronic device which needs to be executed during the installation of each electronic device. At the beginning of each day, electronic device which handles cash transactions must register the amount of cash in the deposit and only then it should start to issue invoices. Each invoice must be registered to the fiscalization service and upon successful registration the invoice is assigned with a FIC which is printed on the invoice together with other mandatory invoice elements. In case that the invoice needs to be corrected, new corrective invoice is issued with a reference to the invoice which needs to be corrected. During the day, electronic cash devices for cash payments should have the possibility for the taxpayer to see the current cash balance (it is recommended to do it when the operators of electronic cash device changes) as well as notify the Tax Administration through fiscalization service of any withdrawal of cash in the cash register. Each of this operation is explained in its chapter together with the list of elements of the exchanged messages.

Message sent by the taxpayer to CIS is the request message to which CIS replies by sending the response message. In case of an error, the error message is sent in the response with its structure. Request and response messages (except for the error message) all have the following parts: header (general info about the message), data (data specific for the operation), signature (digital signature signed by the person who is sending the message which provides the identity of the sender and info to verify that the data of the message is not changed). Signature is explained in chapter **Pogreška! Izvor reference nije pronađen.**.

## INTERFACE VERSIONING

Versioning of the fiscalization service will be based on semantic versioning schema. Each version has a version number assigned expressed as “MAJOR.MINOR.PATCH” each of which are integers incremented according to these rules:

* MAJOR version is increased when there are incompatible API changes. New interface will be provided, and old interface will remain for some period. Clients are expected to upgrade to new version as described in release notes of the new version.
* MINOR version is increased when a functionality is added in a backwards-compatible manner. Current interface remains compatible with current clients, but new functionalities are added which can or should be used. Clients are expected to upgrade to new version as described in release notes of the new version.
* PATCH version is increased when there are backwards-compatible bug fixes. Current interface remains the same.

Service endpoint will have a context suffix -vMAJOR, e.g. /FiscalizationService-v1. This means that at one moment there might be several active service endpoints with different MAJOR versions but each of them will always have the latest MINOR and PATCH versions.

## DATA MESSAGE CODING

All items in all data messages will only use selected characters encoded as a single byte in a standard decimal ASCII character set. The allowed decimal codes are 9, 10, 13, or 32 to 126.

UTF-8 must be used for encoding the data messages as XML documents, i.e. first line of the XML SOAP envelope will always be:

*<?xml version="1.0" encoding="UTF-8"?>*

All XML elements of the fiscalization service are part of the same namespace, referenced in the Web service definition (WSDL).

The data format mask for individual items, which is listed along with their detailed description below, is a regular expression in the sense of the XML Schema, which defines the required syntax of the given item.

## DATA MESSAGE STRUCTURE

All types of data messages have a common basic data format based on the SOAP 1.1 (Simple Object Access Protocol) protocol, i.e. application XML data structures are inserted into the body of the SOAP envelope. Unlike SOAP envelope header which remains empty.

Every request and response data message shall be signed with a private key belonging to the issuer or fiscalization service respectably. Exception to that rule are error messages (described in the chapter **Pogreška! Izvor reference nije pronađen.**) which are not signed by the fiscalization service.

Digital signature is calculated only for the data message that resides inside SOAP envelope body element and is incorporated inside that data message as a envelop signature XML element.

## CONTROLS

There are two types of controls:

* Mandatory controls (in real time)
* Additional controls (during back-up verification)

Mandatory controls shall be performed by CIS system in all methods in real time. In case that the control is not passed, an error message will be returned with error code defined here. The mandatory controls include the following:

|  |  |  |
| --- | --- | --- |
| **Control name** | **Control description** | **Error code** |
| XML format | XML format must be valid | 0.20 |
| Data size check | Size must not be greater than 150kB | 1 |
| XML structure check | Checks the individual recorded electronic messages on invoice data in XML against the XSD schema. XSD schema contains the correct definition of the data structure and formats for individual items of data, and checks the existence of specific items. | 11 |
| Certificate validation | Checks if the certificate is expired.  Checks if the certificate was issued by the authorized body.  Checks that the certificate ID number is competent with the issuer's identifying number (tax number) in XML message.  Checks that the certificate is not listed in the List of withdrawn certificates (CRL or OCSP list). | 34, 35, 36, 37, 38, 39 |
| Electronic signature validation | Checks that message hash calculated by the CIS is aligned with the hash in the message.  Checks that the signature is competent with the hash function of the message and the public certificate key. | 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34 |
| Client time is different | Client time is different than the server time more minutes than it is allowed. | 2 |
| Remaining controls | Remaining controls shown in Functional specification. |  |

Table 7

Additional controls are not performed at the moment of registration of the invoice but are instead postponed for later processing of the invoices. Errors detected here will be available to taxpayer over central invoice platform and to tax officials through central platform.

## ENU REGISTRATION

Each electronic device must be registered on CIS in order to receive the code which represents that electronic cash device. This code is used for identification of electronic device in messages which are exchanged between CIS and the electronic device. This registration must be done only once when the electronic device is installed in the business premise where it is used.

Before this, taxpayer needs to be registered in Tax Administration in active Registry of fiscalization obligors. Taxpayer must also register the business premise in which the ENU is located prior to registration of the ENU.

If this registration is done for the same internal identification of the ENU and business unit code, CIS will update only “valid to” field information with new data provided in the request message and return same ENU code. Deregistration is possible by providing current date as value for “valid to” attribute field date field in the RegisterTCRRequest message.

### REGISTER ENU REQUEST MESSAGE

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | | | **XML tag** | **Field type** | **Occurence [min, max]** | **Description** |
| **Register request message for ENU registration** | | | RegisterTCRRequest | Element | [1, 1] | Root element representing ENU registration request. |
|  | Identifier | | Id | Attribute | [1, 1] | Attribute used for signature creation and checking. |
|  | Version | | Version | Attribute | [1, 1] | Attributes used for checking the compatibility with the XSD schema. For this version value is “1”. |
|  | Header | | Header | Element | [1, 1] | XML element representing the header. |
|  |  | Universal unique identifier | UUID | Attribute | [1, 1] | Message identifier. |
|  |  | Sending date and time | SendDateTime | Attribute | [1, 1] | Date and time of message sending. |
|  | ENU | | TCR | Element | [1, 1] | XML element representing one ENU registration message. |
|  |  | Issuer PIB | IssuerTIN | Attribute | [1, 1] | Taxpayer PIB. |
|  |  | Business unit code | BusinUnitCode | Attribute | [1, 1] | Business unit code. |
|  |  | Internal code | TCRIntId | Attribute | [1, 1] | Internal ENU identifier (especially if there are more ENUs in one business unit) |
|  |  | Software code | SoftCode | Attribute | [0, 1] | Code of the software that the ENU is using. |
|  |  | Maintainer code | MaintainerCode | Attribute | [0, 1] | Software maintainer code. |
|  |  | Valid from | ValidFrom | Attribute | [0, 1] | Date since ENU is used. |
|  |  | Valid to | ValidTo | Attribute | [0, 1] | Date until ENU is used. |
|  |  | Type | Type | Attribute | [0, 1] | ENU type. |
|  | Signature | | Signature | Element | [1, 1] | XML element with the signature. |

Table 8

#### Header

Element representing the header of the request data message.

#### Header: Universal unique identifier

Element generated by the ENU. It uniquely identifies the request message sent from ENU to CIS. UUID should be constructed according to the RFC4122 version 4.

|  |  |
| --- | --- |
| Data type | String |
| Length | 36 chars |
| Pattern | [0-9a-FA-F]{8}–[0-9a-FA-F]{4}–[1-5][0-9a-FA-F]{3}–[89abAB][0-9a-FA-F]{3}–[0-9a-FA-F]{12} |
| Example | 58e0a7d7-eebc-41d8-9669-0800200c9a66  58E0A7D7-EEBC-41D8-9669-0800200C9A66 |

Table 9

#### Header: Sending date and time

Element represents date and time of sending the request message to the CIS. Date and time should be in ISO 8601 format.

|  |  |
| --- | --- |
| Data type | DateTime |
| Pattern | [0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}([+-][0-9]{2}:[0-9]{2}|Z) |
| Example | 2019-01-24T22:00:58+01:00  2019-01-24T22:00:58-01:00 |

Table 10

#### ENU

Element representing a single ENU registration request.

#### ENU: Issuer PIB

Element representing the tax identification number of the issuer.

|  |  |
| --- | --- |
| Data type | String |
| Length | 8 numeric chars for legal persons, 13 numeric chars for natural persons |
| Pattern | [0-9]{13}|[0-9]{8} |
| Example | 02657597 |

Table 11

#### ENU: Business unit code

Code of the business unit where the ENU is located.

|  |  |
| --- | --- |
| Data type | String |
| Length | 10 chars |
| Pattern | [a-z]{2}[0-9]{3}[a-Z]{2}[0-9]{3} |
| Example | ab123ab123 |

Table 12

#### ENU: Internal code

Attribute that represents the internal identification of the ENU.

|  |  |
| --- | --- |
| Data type | String |
| Length | 50 chars |
| Example | 2 |

Table 13

#### ENU: Software code

Code of the software used for invoice issuing.

|  |  |
| --- | --- |
| Data type | String |
| Length | 10 chars |
| Pattern | [a-z]{2}[0-9]{3}[a-Z]{2}[0-9]{3} |
| Example | ab123ab123 |

Table 14

#### ENU: Maintainer code

Code of the software maintainer.

|  |  |
| --- | --- |
| Data type | String |
| Length | 3 numeric chars |
| Pattern | [a-z]{2}[0-9]{3}[a-Z]{2}[0-9]{3} |
| Example | mm123mm123 |

Table 15

#### ENU: Valid from

Element representing the date since the ENU is valid or in use.

|  |  |
| --- | --- |
| Data type | Date |
| Length | 10 chars |
| Pattern | [[0-9]{4}-[0-9]{2}-[0-9]{2} |
| Example | 2019-01-24 |

Table 16

#### ENU: Valid to

Element representing the date until the ENU is valid or in use.

|  |  |
| --- | --- |
| Data type | Date |
| Length | 10 chars |
| Pattern | [[0-9]{4}-[0-9]{2}-[0-9]{2} |
| Example | 2019-01-24 |

Table 17

#### ENU: Type

This attribute shows the type of the cash device.

|  |  |
| --- | --- |
| Data type | String |
| Value | ENUMERATION |
| Example | REGULAR |

Table 18

Table below shows the list of available values for Type attribute.

|  |  |
| --- | --- |
| Value | Description |
| REGULAR | Standard ENU |
| VENDING | Self-vending machine |

Table 19

#### Signature

XML element stores enveloped digital signature described in the chapter **Pogreška! Izvor reference nije pronađen.**.

### REGISTER ENU RESPONSE MESSAGE

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Name** | | | **XML tag** | | **Field type** | **Occurance [min, max]** | **Description** |
| **Register response message for ENU registration** | | | | RegisterTCRResponse | Element | [1, 1] | Root element representing ENU registration response. |
|  | Identifier | | | Id | Attribute | [1, 1] | Attribute used for signature creation and checking. |
|  | Version | | | Version | Attribute | [1, 1] | Attributes used for checking the compatibility with the XSD schema. For this version value is “1”. |
|  | Header | | | Header | Element | [1, 1] | XML element representing the header. |
|  |  | Universal unique identifier | | UUID | Attribute | [1, 1] | Message identifier. |
|  |  | Universal unique request identifier | | RequestUUID | Attribute | [1,1] | Message request identifier. |
|  |  | Date and time of sending | | SendDateTime | Attribute | [1, 1] | Date and time of message sending |
|  | ENU code | | | TCRCode | Element | [1, 1] | ENU code generated by the CIS |
|  | Signature | | | Signature | Element | [1, 1] | XML element with the digital signature |

Table 20

#### Header

Element representing the header of the response data message.

#### Header: Universal unique identifier

Element generated by the CIS. It uniquely identifies the response message sent from CIS to ENU. UUID should be constructed according to the RFC4122 version 4.

|  |  |
| --- | --- |
| Data type | String |
| Length | 36 chars |
| Pattern | [0-9a-FA-F]{8}-[0-9a-FA-F]{4}-[1-5][0-9a-FA-F]{3}-[89abAB][0 -9a-FA-F]{3}-[0-9a-FA-F]{12} |
| Example | 58e0a7d7-eebc-41d8-9669-0800200c9a66  58E0A7D7-EEBC-41D8-9669-0800200C9A66 |

Table 21

#### Header: Universal unique request identifier

Element generated by the ENU and referenced by the CIS. It uniquely identifies the request message for which the response message was sent to the ENU. UUID should be constructed according to the RFC4122 version 4.

|  |  |
| --- | --- |
| Data type | String |
| Length | 36 chars |
| Pattern | [0-9a-FA-F]{8}-[0-9a-FA-F]{4}-[1-5][0-9a-FA-F]{3}-[89abAB][0-9a-FA-F]{3}-[0-9a-FA-F]{12} |
| Example | 58e0a7d7-eebc-41d8-9669-0800200c9a66  58E0A7D7-EEBC-41D8-9669-0800200C9A66 |

Table 22

#### Header: Date and time of sending

Date and time of sending the response. Both are supposed to be in ISO 8601 format.

|  |  |
| --- | --- |
| Data type | DateTime |
| Pattern | [0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}([+-][0-9]{2}:[0-9]{2}|Z) |
| Example | 2019-01-24T22: 00: 58 + 01: 00  2019-01-24T22: 00: 58-01: 00 |

Table 23

#### ENU code

ENU code, generated by the CIS.

|  |  |
| --- | --- |
| Data type | String |
| Length | 10 chars |
| Pattern | [a-z]{2}[0-9]{3}[a-Z]{2}[0-9]{3} |
| Example | ab123ab123 |

Table 24

#### Signature

XML element stores enveloped digital signature described in the chapter **Pogreška! Izvor reference nije pronađen.**.

### MANDATORY CONTROLS

Mandatory controls shall be performed by CIP system in process of registering the cash deposit of ENU. In case that the control is not passed, an error message will be returned with an error code.

Upon identifying a critical error, CIS will return an error data message containing the error’s numeric code and its text description (see chapter **Pogreška! Izvor reference nije pronađen.**). When errors which the system can interpret as a cyber-attack are identified, the system does not send any response to the client (the taxpayer’s ENU).

The mandatory controls include all the controls from chapter **Pogreška! Izvor reference nije pronađen.** and the following:

|  |  |  |
| --- | --- | --- |
| Control name | Control description | Error code |
| „Valid from“ in past | „Valid from“ cannot be in past. | 45 |
| „Valid to“ in past | „Valid to“ cannot be in past.. | 46 |
| „Valid to“ before „Valid from“ | „Valid to“ cannot be before „Valid from“. | 47 |
| ENU update active | For the active ENU only possible updates are for „Valid to“ and „Software maintainer code“ fields. | 48 |
| Deactivated ENU update | Deactivated ENU cannot be updated. | 57 |
| Registered taxpayer | Issuer unique identifying number is not related to the active taxpayer in Register of taxpayers. | 52 |
| Registered business unit | Business unit code is not related to the taxpayer's business unit. | 41 |
| Registered software | Software code is not related to the active software version. | 42 |
| Registered maintainer | Maintainer code is not related to the taxpayer. | 43 |

Table 25

### ERROR MESSAGE

Error message is defined in chapter **Pogreška! Izvor reference nije pronađen.**.

### XML EXAMPLES

#### XML request

<SOAP-ENV:Envelope xmlns:SOAP-ENV=**"http://schemas.xmlsoap.org/soap/envelope/"**>

<SOAP-ENV:Header/>

<SOAP-ENV:Body>

<RegisterTCRRequest xmlns=**"https://efi.tax.gov.me/fs/schema"**

xmlns:ns2=**"http://www.w3.org/2000/09/xmldsig#"** Id=**"Request"** Version=**"1"**>

<Header SendDateTime=**"2019-12-02T10:57:01+01:00"** UUID=**"78b37523-3677-416a-8bc0-e0dd77296fc7"**/>

<TCR BusinUnitCode**="ab123ab123"** IssuerTIN=**"02657597"** MaintainerCode=**"123"** SoftCode=**" ab123ab123"** TCRIntID=**"1"** ValidFrom**="2019-12-05"** Type**="REGULAR"**/>

<Signature xmlns=**"http://www.w3.org/2000/09/xmldsig#"**>

<SignedInfo>

<CanonicalizationMethod Algorithm=**"http://www.w3.org/2001/10/xml-exc-c14n#"**/>

<SignatureMethod Algorithm=**"http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"**/>

<Reference URI=**"#Request"**>

<Transforms>

<Transform Algorithm=**"http://www.w3.org/2000/09/xmldsig#enveloped-signature"**/>

<Transform Algorithm=**"http://www.w3.org/2001/10/xml-exc-c14n#"**/>

</Transforms>

<DigestMethod Algorithm=**"http://www.w3.org/2001/04/xmlenc#sha256"**/>

<DigestValue>**0ilvSabbyg8..........GzADJ0DO8=**</DigestValue>

</Reference>

</SignedInfo>

<SignatureValue>**tpGg48z..........0HmXIVOA==**</SignatureValue>

<KeyInfo>

<X509Data>

<X509Certificate>**MIIFYDCCB..........Pifz0UlKJAanmqN3**</X509Certificate>

</X509Data>

</KeyInfo>

</Signature>

</RegisterTCRRequest>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

#### XML response

<SOAP-ENV:Envelope xmlns:SOAP-ENV=**"http://schemas.xmlsoap.org/soap/envelope/"**>

<SOAP-ENV:Header/>

<SOAP-ENV:Body>

<ns2:RegisterTCRResponse Id=**"Response" Version="1"** xmlns:ns2=**" https://efi.tax.gov.me/fs/schema "** xmlns:ns3=**"http://www.w3.org/2000/09/xmldsig#"**>

<ns2:Header RequestUUID=**"07986698-1502-4150-9520-1918a5bbbf49"** SendDateTime=**"2019-11-27T09:49:22+01:00"** UUID=**"12839912-75b6-4cd4-ba2d-fca3e6395226"**/>

<ns2:TCRCode>**ab123ab123**</ns2:TCRCode>

<Signature xmlns=**"http://www.w3.org/2000/09/xmldsig#"**>

<SignedInfo>

<CanonicalizationMethod Algorithm=**"http://www.w3.org/2001/10/xml-exc-c14n#"**/>

<SignatureMethod Algorithm=**"http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"**/>

<Reference URI=**"#Response"**>

<Transforms>

<Transform Algorithm=**"http://www.w3.org/2000/09/xmldsig#enveloped-signature"**/>

<Transform Algorithm=**"http://www.w3.org/2001/10/xml-exc-c14n#"**/>

</Transforms>

<DigestMethod Algorithm=**"http://www.w3.org/2001/04/xmlenc#sha256"**/>

<DigestValue>**p+TSatU9..........+yD8ff87U=**</DigestValue>

</Reference>

</SignedInfo>

<SignatureValue>**kFXWTPybI..........W6kKA3ojg==**</SignatureValue>

<KeyInfo>

<X509Data>

<X509Certificate>**MIIFRjCCBC..........gpNT2r23YQ==**</X509Certificate>

</X509Data>

</KeyInfo>

</Signature>

</ns2:RegisterTCRResponse>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

## CASH DEPOSIT REGISTRATION

Every day before registering the first cash invoice of the day on CIS, each ENU for handling cash transactions must register the initial amount of cash deposit (Operation INITIAL). Registration of the initial cash deposit is possible only once during the business day, whereby changes can be made until the moment of issuing the first invoice. The final value of the deposit is the one that the taxpayer registered as the last before issuing the first invoice.

During the day, the operator can take cash from the ENU and each of these actions must be registered (Operation WITHDRAW). These changes must also be submitted to the Tax Administration via the service at the moment when they occur, ie when they withdraw a certain amount of cash from ENU during the working day.

In special cases defined by the Law (when there is failure of internet connection or if the taxpayer operates in the area without internet connection), information about opening deposit or withdrawal must be stored in ENU memory. If there is failure of ENU, data of the deposit/withdrawal is written in the verified invoice book and submitted to the Tax Administration later together with all invoices that are going to be fiscalized through that process.

The open cash deposit can be 0.00.

### REGISTER CASH DEPOSIT REQUEST MESSAGE

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Naziv** | | | **XML tag** | **Vrsta polja** | **Pojava [minimalno, maksimalno]** | **Opis** |
| **Register request message for cash deposit** | | | RegisterCashDepositRequest | Element | [1, 1] | Root element representing the cash deposit response. |
|  | Identifier | | Id | Attribute | [1, 1] | Attribute used for signature creation and checking. |
|  | Version | | Version | Attribute | [1, 1] | Attributes used for checking the compatibility with the XSD schema. For this version value is “1”. |
|  | Header | | Header | Element | [1, 1] | XML element representing the header. |
|  |  | Universal unique identifier | UUID | Attribute | [1, 1] | UUID generated by ENU for every data message sent to the CIS. |
|  |  | Date and time of sending | SendDateTime | Attribute | [1, 1] | Date and time of message sending. |
|  |  | Subsequent delivery type | SubseqDelivType | Element | [0, 1] | Message is delivered subsequently. |
|  | Cash deposit | | CashDeposit | Element | [1, 1] | XML element representing a single request for cash deposit. |
|  |  | Deposit change date and time | ChangeDateTime | Attribute | [1, 1] | Date and time of deposit change. |
|  |  | Cash deposit operation | Operation | Attribute | [1,1] | Operation done on ENU. |
|  |  | Cash amount | CashAmt | Attribute | [1, 1] | Cash deposit amount. |
|  |  | ENU code | TCRCode | Attribute | [1, 1] | ENU code on which the deposit is registered. |
|  |  | PIB | IssuerTIN | Attribute | [1, 1] | JMB/PIB of the taxpayer. |
|  | Signature | | Signature | Element | [1, 1] | XML element with the digital signature. |

Table 26

#### Header

Element representing the header of the request data message.

#### Header: Universal unique identifier

Element generated by the ENU. It uniquely identifies the request message sent from ENU to CIS. UUID should be constructed according to the RFC4122 version 4.

|  |  |
| --- | --- |
| Data type | String |
| Length | 36 chars |
| Pattern | [0-9a-FA-F]{8}-[0-9a-FA-F]{4}-[1-5][0-9a-FA-F]{3}-[89abAB][0-9a-FA-F]{3}-[0-9a-FA-F]{12} |
| Example | 58e0a7d7-eebc-41d8-9669-0800200c9a66  58E0A7D7-EEBC-41D8-9669-0800200C9A66 |

Table 27

#### Header: Sending date and time

Element represents date and time of sending the request message to the CIS. Date and time should be in ISO 8601 format.

|  |  |
| --- | --- |
| Data type | Date and time |
| Pattern | [0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}([+-][0-9]{2}:[0-9]{2}|Z) |
| Example | 2019-01-24T22:00:58+01:00  2019-01-24T22:00:58-01:00 |

Table 28

#### Header: Subsequent delivery type

Element that says what is the type of subsequent delivery. For example, when the issuer sent the message to the CIS, but he did not receive the answer, he can send another message as before, but only with a flag with “Subsequent delivery” note. This flag indicates to CIS that the issuer sent another message.

|  |  |
| --- | --- |
| Data type | String |
| Value | ENUMERATION |
| Example | NOINTERNET |

Table 29

Table below shows the list of available values for Subsequent delivery type attribute.

|  |  |
| --- | --- |
| Value | Description |
| NOINTERNET | If ENU works in area without internet |
| BOUNDBOOK | ENU is not working and no message can be created |
| SERVICE | Fiscalization service problem |
| TECHNICALERROR | Technical error |
| BUSINESSNEED | Subsequent sending conditioned by the way of doing business |

Table 30

#### Cash deposit

Element representing one cash deposit registration.

#### Cash deposit: Deposit change date and time

Element representing date and time of the change of the deposit.

|  |  |
| --- | --- |
| Data type | Date and time |
| Pattern | [0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}([+-][0-9]{2}:[0-9]{2}|Z) |
| Example | 2019-01-24T22:00:58+01:00  2019-01-24T22:00:58-01:00 |

Table 31

#### Cash deposit: Cash deposit operation

This is an XML element representing the operation done on the ENU.

|  |  |
| --- | --- |
| Data type | String |
| Value | ENUMERATION |
| Example | INITIAL |

Table 32

Table below shows the list of available values for Cash deposit operation.

|  |  |
| --- | --- |
| Value | Description |
| INITIAL | Initial ENU cash amount |
| WITHDRAW | Cash amount withdrawn from the ENU |

Table 33

#### Cash deposit: cash amount

Element representing the amount of cash in the ENU.

|  |  |
| --- | --- |
| Data type | Decimal |
| Pattern | ([1-9][0-9]\*|0)?.\[0-9]{2}|0 |
| Example | 212.12 |

Table 34

#### Cash deposit: ENU code

Unique code of the ENU concerned.

|  |  |
| --- | --- |
| Data type | String |
| Length | 10 chars |
| Pattern | [a-z]{2}[0-9]{3}[a-Z]{2}[0-9]{3} |
| Example | ab123ab123 |

Table 35

#### Cash deposit: Issuer PIB

Issuer Tax identification number (JMB/PIB).

|  |  |
| --- | --- |
| Data type | String |
| Length | 8 numeric chars for legal persons, 13 numeric chars for natural persons |
| Pattern | [0-9]{13}[0-9]{8} |
| Example | 72001008 |

Table 36

#### Signature

XML element stores enveloped digital signature described in the chapter **Pogreška! Izvor reference nije pronađen.**.

### REGISTER CASH DEPOSIT RESPONSE MESSAGE

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | | | **XML tag** | **Field type** | **Occurrence [min, max]** | **Description** |
| **Register response message for cash deposit** | | | RegisterCashDepositResponse | Element | [1, 1] | Root element representing the cash deposit request. |
|  | Identifier | | Id | Attribute | [1, 1] | Attribute used for signature creation and checking. |
|  | Version | | Version | Attribute | [1, 1] | Attributes used for checking the compatibility with the XSD schema. For this version value is “1”. |
|  | Header | | Header | Element | [1, 1] | XML element representing the header. |
|  |  | Universal unique identifier | UUID | Attribute | [1, 1] | Message identifier |
|  |  | Universal unique request identifier | RequestUUID | Attribute | [1, 1] | UUID of the request message. |
|  |  | Date and time of sending | SendDateTime | Attribute | [1, 1] | Date and time of message sending to the CIS. |
|  | Cash deposit fiscalization code | | FCDC | Element | [1, 1] | Cash deposit fiscalization code generated by CIS. |
|  | Signature | | Signature | Element | [1, 1] | XML element with the digital signature. |

Table 37

#### Header

Element representing the header of the request data message.

#### Header: Universal unique identifier

Element generated by the ENU. It uniquely identifies the request message sent from ENU to CIS. UUID should be constructed according to the RFC4122 version 4.

|  |  |
| --- | --- |
| Data type | String |
| Length | 36 chars |
| Pattern | [0-9a-FA-F]{8}–[0-9a-FA-F]{4}–[1-5][0-9a-FA-F]{3}–[89abAB][0-9a-FA-F]{3}–[0-9a-FA-F]{12} |
| Example | 58e0a7d7-eebc-41d8-9669-0800200c9a66  58E0A7D7-EEBC-41D8-9669-0800200C9A66 |

Table 38

#### Header: Universal unique request identifier

Element generated by the ENU and referenced by the CIS. It uniquely identifies the request message for which the response message was sent to the ENU. UUID should be constructed according to the RFC4122 version 4.

|  |  |
| --- | --- |
| Data type | String |
| Length | 36 chars |
| Pattern | [0-9a-FA-F]{8}–[0-9a-FA-F]{4}–[1-5][0-9a-FA-F]{3}–[89abAB][0-9a-FA-F]{3}–[0-9a-FA-F]{12} |
| Example | 58e0a7d7-eebc-41d8-9669-0800200c9a66  58E0A7D7-EEBC-41D8-9669-0800200C9A66 |

Table 39

#### Header: Date and time of sending

|  |  |
| --- | --- |
| Data type | Date and time |
| Pattern | [0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}([+-][0-9]{2}:[0-9]{2}|Z) |
| Example | 2019-01-24T22:00:58+01:00  2019-01-24T22:00:58-01:00 |

Table 40

#### Cash deposit fiscalization code

Cash deposit fiscalization code. Unique code generated by the CIS for each successful cash deposit registration.

|  |  |
| --- | --- |
| Data type | String |
| Length | 36 chars |
| Pattern | [0-9a-FA-F]{8}–[0-9a-FA-F]{4}–[1-5][0-9a-FA-F]{3}–[89abAB][0-9a-FA-F]{3}–[0-9a-FA-F]{12} |
| Example | 58e0a7d7-eebc-41d8-9669-0800200c9a66  58E0A7D7-EEBC-41D8-9669-0800200C9A66 |

Table 41

#### Signature

XML element stores enveloped digital signature described in the chapter **Pogreška! Izvor reference nije pronađen.**.

### MANDATORY CONTROLS

Mandatory controls shall be performed by CIS system in process of registering the cash deposit of ENU. In case that the control is not passed, an error message will be returned with an error code.

Upon identifying a critical error, CIS will return an error data message containing the error’s numeric code and its text description (see chapter **Pogreška! Izvor reference nije pronađen.**). When errors which the system can interpret as a cyber-attack are identified, the system does not send any response to the client (the taxpayer’s ENU).

The mandatory controls include all the controls from chapter **Pogreška! Izvor reference nije pronađen.** and the following:

|  |  |  |
| --- | --- | --- |
| Control name | Control description | Error code |
| Check of change in date and time | Change in date and time differs from the time recorder in CIS for more than what is allowed. | 49 |
| Amount check when withdrawing | Cash amount is zero and the operation equals WITHDRAW | 51 |
| Registered taxpayer check | Issuer personal identification number is not as same as active taxpayer in Register of taxpayers. | 52 |
| Registered ENU check | ENU code is not the same as registered or active ENU code or does not belong to the issuer in question. | 53 |
| Initial deposit registration check | If the cash deposit is already registered for the current day with operation INITIAL. | 56 |

Table 42

### ERROR MESSAGE

Error message is defined in chapter **Pogreška! Izvor reference nije pronađen.**.

### XML EXAMPLE

#### XML request

<SOAP-ENV:Envelope xmlns:SOAP-ENV=**http://schemas.xmlsoap.org/soap/envelope/"**>

<SOAP-ENV:Header/>

<SOAP-ENV:Body>

<RegisterCashDepositRequest xmlns=**" https://efi.tax.gov.me/fs/schema"** xmlns:ns2=**"http://www.w3.org/2000/09/xmldsig#"** Id=**"Request"** Version=**"1"**>

<Header SendDateTime=**"2019-12-05T14:35:00+01:00"** UUID=**"3389b9c4-bb24-4673-b952-456e451cd3c3"**/>

<CashDeposit CashAmt=**"2000.00"** ChangeDateTime=**"2019-12-05T14:35:00+01:00"** IssuerTIN=**"EXAMPLE PIB-A"** Operation=**"INITIAL"** TCRCode=**"KOD BLAGAJNE"**/>

<Signature xmlns=**"http://www.w3.org/2000/09/xmldsig#"**>

<SignedInfo>

<CanonicalizationMethod Algorithm=**"http://www.w3.org/2001/10/xml-exc-c14n#"**/>

<SignatureMethod Algorithm=**"http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"**/>

<Reference URI=**"#Request"**>

<Transforms>

<Transform Algorithm=**"http://www.w3.org/2000/09/xmldsig#enveloped-signature"**/>

<Transform Algorithm=**"http://www.w3.org/2001/10/xml-exc-c14n#"**/>

</Transforms>

<DigestMethod Algorithm=**"http://www.w3.org/2001/04/xmlenc#sha256"**/>

<DigestValue>**PTmGp/uQH.......... LR+4IH2/hODo=**</DigestValue>

</Reference>

</SignedInfo>

<SignatureValue>**PDV1uTMr5..........SspzFpjYkEA==**</SignatureValue>

<KeyInfo>

<X509Data>

<X509Certificate>**MIIFYDCCB..........** **..........Pifz0UlKJAanmqN3**</X509Certificate>

</X509Data>

</KeyInfo>

</Signature>

</RegisterCashDepositRequest>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

#### XML response

<SOAP-ENV:Envelope xmlns:SOAP-ENV=**"http://schemas.xmlsoap.org/soap/envelope/"**>

<SOAP-ENV:Header/>

<SOAP-ENV:Body>

<ns2:RegisterCashDepositResponse Id=**"Response" Version="1"** xmlns:ns2=**" https://efi.tax.gov.me/fs/schema "** xmlns:ns3=**"http://www.w3.org/2000/09/xmldsig#"**>

<ns2:Header RequestUUID=**"5b685bba-fe1e-405c-ac8c-2243502dc9db"** SendDateTime=**"2019-11-27T09:44:14+01:00"** UUID=**"ed3cbe5c-08ad-44e2-a6e1-a811e3c80137"**/>

<ns2:FCDC>**c8a197f1-b7dc-4e82-8729-4a506aecf9b9**</ns2:FCDC>

<Signature xmlns=**"http://www.w3.org/2000/09/xmldsig#"**>

<SignedInfo>

<CanonicalizationMethod Algorithm=**"http://www.w3.org/2001/10/xml-exc-c14n#"**/>

<SignatureMethod Algorithm=**"http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"**/>

<Reference URI=**"#Response"**>

<Transforms>

<Transform Algorithm=**"http://www.w3.org/2000/09/xmldsig#enveloped-signature"**/>

<Transform Algorithm=**"http://www.w3.org/2001/10/xml-exc-c14n#"**/>

</Transforms>

<DigestMethod Algorithm=**"http://www.w3.org/2001/04/xmlenc#sha256"**/>

<DigestValue>**giebs7Pyu..........rs+xz0iS3f0=**</DigestValue>

</Reference>

</SignedInfo>

<SignatureValue>**BwiADbuZFaa..........fd2Uo3LEiClw==**</SignatureValue>

<KeyInfo>

<X509Data>

<X509Certificate>**MIIFRjCCBC6g..........JfgpNT2r23YQ==**</X509Certificate>

</X509Data>

</KeyInfo>

</Signature>

</ns2:RegisterCashDepositResponse>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

## INVOICE REGISTRATION AND FISCALIZATION

Issuer of the invoice is obliged to deliver information on each invoice he issues. Information has to be delivered at the moment of issuing. Exceptionally, it can be delivered afterwards (as stipulated in the Law).

Data exchange process starts at the moment when the issuer is about to issue an invoice to the customer. The ENU prepares invoice data and based on that data it creates IKOF. After that it prepares XML invoice request message and signs it electronically by its certificate private key (using a certificate that was issued to issuer by CIS with purpose being the implementation of fiscalization). After that the 1-way TLS communication is started and once successful it calls the service.

Central information system receives and processes request message. If the request is successfully processed, central information system prepares XML message that contains JIKR, which is unique for every invoice, signs it electronically with its certificate, and sends it back to the ENU.

ENU receives answer message and checks its electronic signature. After that, cashier (operator) issues the invoice and hands it to the buyer.

Corrective invoice is a special type of invoice which contains a reference to the original invoice and is issued in order to change some data in the original invoice, e.g. some items from the original invoice should be removed because they are returned from the buyer to the seller.

If there were errors during the operation (invalid XML, invalid certificate or similar), central information system shows the error as XML message. If that is the case, there is no IKOF so issuer will issue the invoice without JIKR. The invoice issuing process must not be halted because of the error, but the issuer is obliged to correct the error and deliver it after he receives the correct message.

In all situations when issuer does not get JIKR for invoice he issued (loss of Internet connection, computer breakdown, central information system unavailability or similar), he is obliged to make another invoice request. Invoice is found to be properly sent and reported to the CIS once issuer gets JIKR for it.

In cases when there are invoices without JIKR, those should be sent again later (and in timeframe defined in laws), as invoices processed at the moment have advantage over invoices issued prior. Invoices without JIKR should be delivered when traffic load is smaller or immediately and automatically when Internet connection becomes available again.

When the taxpayer is operating in the area without internet connection, he can export invoice registration request into special file format which can be then delivered through the Central invoice platform with bulk upload of invoice or brought to the Tax Administration office.

Maximum time-out for machine to wait for the answer that contains JIKR is set by issuer himself. Issuer needs to check the Internet connection quality and time needed for issuing one receipt so that will not affect his business. When calculating maximum time-out, issuer should count in additional two seconds (time needed for request to come in and get out of process).

Invoice registration can be verified by online application. Each invoice contains a QR-code which contains a link to the web page which displays information about the invoice, if it was successfully registered. Details are explained in subchapter 3.7.7.

CIS processes the registration of the invoice in the following way:

* TLS communication is initiated between the invoice issuer and CIS
  + In case that certificate provided by the invoice issuer is invalid, communication protocol fails
* CIS checks the size of the message
  + If the message is larger than allowed, an error message is sent in the response and no further processing is done.
* Further controls are performed. For each passed control, process continues to the next control. If the control fails, no more controls are performed, and error message is prepared. Following controls are performed:
  + SOAP message is a valid XML document
  + SOAP message is structured according to the defined schema
  + Certificate in the signature of the message is checked
  + Signature of the message is checked
  + IKOF is verified
  + Date and time written in the message is checked
  + Is issuer in the VAT register
* If one of the controls failed, an error message will be sent in the reply to the request message with the error code related to the nature of the error.
* If all of the controls have passed successfully, register invoice response message will be sent.
  + JIKR code is generated.
  + Response message is prepared with JIKR included.
  + Response message is signed, and signature is put into the message.
* Invoice data is inserted into the database.
  + Data includes request SOAP message, response SOAP message (or error message).
* Prepared response message is sent back to the taxpayer.

### REGISTER INVOICE REQUEST MESSAGE

This is request message that the invoice issuer sent to the CIS.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Name** | | | | | | | | | | | | | | | | | | | | | | **XML tag** | **Field type** | **Occurrence [min, max]** | **Description** | |
| **Register request message for register invoice** | | | | | | | | | | | | | | | | | | | | | | RegisterInvoiceRequest | Osnovno | [1, 1] | Root XML element representing register invoice message. | |
|  | Identifier | | | | | | | | | | | | | | | | | | | | | Id | Attribute | [1, 1] | Attribute used for signature creation and verification. Fixed value “Request”. | |
|  | Version | | | | | | | | | | | | | | | | | | | | | Version | Attribute | [1, 1] | Attribute used to specify compliance with XSD schema. For this version fixed value is “3”. | |
|  | Header | | | | | | | | | | | | | | | | | | | | | Header | Element | [1, 1] | XML element representing header of the invoice containing data about the message (request) sent. | |
|  |  | Universal unique identifier | | | | | | | | | | | | | | | | | | | | UUID | Attribute | [1, 1] | UUID generated by ENU for every register sale data message send to the CIS. | |
|  |  | Date and time of sending | | | | | | | | | | | | | | | | | | | | SendDateTime | Attribute | [1, 1] | Date and time of sending the register invoice data message from ENU to the CIS. | |
|  |  | Subsequent delivery type | | | | | | | | | | | | | | | | | | | | SubseqDelivType | Attribute | [0, 1] | Type of subsequent delivery if message is delivered after invoice issuance. | |
|  | Invoice | | | | | | | | | | | | | | | | | | | | | Invoice | Element | [1, 1] | XML element representing a single invoice. | |
|  |  | Invoice type | | | | | | | | | | | | | | | | | | | | TypeOfInv | Attribute | [1, 1] | Type of the invoice (cash, non-cash). | |
|  |  | Invoice type issued by the buyer | | | | | | | | | | | | | | | | | | | | TypeOfSelfiss | Attribute | [0, 1] | Entered only if invoice is self-issued. | |
|  |  | Simplified invoice | | | | | | | | | | | | | | | | | | | | IsSimplifiedInv | Attribute | [1, 1] | Is invoice simplified | |
|  |  | Issuing date and time | | | | | | | | | | | | | | | | | | | | IssueDateTime | Attribute | [1, 1] | Date and time when the invoice is created and issued at ENU. | |
|  |  | Number | | | | | | | | | | | | | | | | | | | | InvNum | Attribute | [1, 1] | Invoice number. | |
|  |  | Ordinal number | | | | | | | | | | | | | | | | | | | | InvOrdNum | Attribute | [1, 1] | Invoice ordinal number. | |
|  |  | ENU code | | | | | | | | | | | | | | | | | | | | TCRCode | Attribute | [1, 1] | Code of the device that issued the invoice. | |
|  |  | Issuer is in VAT | | | | | | | | | | | | | | | | | | | | IsIssuerInVAT | Attribute | [1, 1] | Issuer is in VAT register | |
|  |  | Amount without VAT | | | | | | | | | | | | | | | | | | | | TaxFreeAmt | Attribute | [0, 1] | The total amount of goods and services delivered when VAT is not charged | |
|  |  | Margin amount | | | | | | | | | | | | | | | | | | | | MarkupAmt | Attribute | [0, 1] | Amount related to special procedure for margin scheme. Refers to Art. 45 and Art. 46 paragraph 1 of the Law on VAT. | |
|  |  | Amount of export of goods/services | | | | | | | | | | | | | | | | | | | | GoodsExAmt | Attribute | [0, 1] | Amount of goods for export from Montenegro. | |
|  |  | Total invoice price without VAT | | | | | | | | | | | | | | | | | | | | TotPriceWoVAT | Attribute | [1, 1] | Total price of the invoice excluding VAT. | |
|  |  | Total VAT amount | | | | | | | | | | | | | | | | | | | | TotVATAmt | Attribute | [0, 1] | Total VAT amount of the invoice.  Mandatory if the issuer is in the VAT system. | |
|  |  | Total price | | | | | | | | | | | | | | | | | | | | TotPrice | Attribute | [1, 1] | Total price of all items including taxes and discounts. | |
|  |  | Operator code | | | | | | | | | | | | | | | | | | | | OperatorCode | Attribute | [1, 1] | Reference to the operator code, who is operating on ENU and issues invoices. | |
|  |  | Business unit code | | | | | | | | | | | | | | | | | | | | BusinUnitCode | Attribute | [1, 1] | Business unit (premise) code. | |
|  |  | Software code | | | | | | | | | | | | | | | | | | | | SoftCode | Attribute | [1, 1] | Software code. | |
|  |  | Invoice issuer code (IKOF) | | | | | | | | | | | | | | | | | | | | IIC | Attribute | [1, 1] | Issuer’s invoice code calculated as MD5 hash from IICSignature attribute. | |
|  |  | IKOF signature | | | | | | | | | | | | | | | | | | | | IICSignature | Attribute | [1, 1] | Signed issuer’s invoice code concatenated parameters. | |
|  |  | Reverse charge | | | | | | | | | | | | | | | | | | | | IsReverseCharge | Attribute | [1, 1] | If true, the buyer is obliged to pay the VAT.  Currently, this option does not exist in the Law of Montenegro, so the value false is always sent for this field. | |
|  |  | Payment deadline | | | | | | | | | | | | | | | | | | | | PayDeadline | Attribute | [0,1] | Last day for payment. | |
|  |  | Paragon block number | | | | | | | | | | | | | | | | | | | | ParagonBlockNum | Attribute | [0,1] | If the invoice was created on a paragon block, the number of that paragon block is entered. | |
|  |  | Tax period | | | | | | | | | | | | | | | | | | | | TaxPeriod | Attribute | [0,1] | Tax period that invoice belongs. | |
|  |  | Corrective invoice | | | | | | | | | | | | | | | | | | | | CorrectiveInv | Element | [0, 1] | XML element groups data for an original invoice that will be corrected with current invoice. | |
|  |  |  | | | | | | | Reference on IKOF | | | | | | | | | | | | | IICRef | Attribute | [1, 1] | IKOF reference on the original invoice. | |
|  |  |  | | | | | | | Issuing date and time | | | | | | | | | | | | | IssueDateTime | Attribute | [1, 1] | Date and time the original invoice is created and issued at ENU. | |
|  |  |  | | | | | | | Type | | | | | | | | | | | | | Type | Attribute | [1, 1] | Type of the corrective invoice. | |
|  |  | Summary invoices with a list of IKOF references | | | | | | | | | | | | | | | | | | | | SumInfIICRefs | Element | [0, 1] | XML element that contains list of IKOFs to which this invoice referred to, e.g. if this is a summary invoice it shall contain a reference to each individual invoice issued and fiscalized before and included in this summary invoice. | |
|  |  |  | | | | | | | | | Summary invoice with an IKOF reference | | | | | | | | | | | SumInvIICRef | Element | [1, 1000] | XML element that contains one IKOF reference, e.g. reference of the invoice that is part of the summary invoice. | |
|  |  |  | | | | | | | | |  | | | | | IKOF | | | | | | IIC | Attribute | [1, 1] | IKOF of the invoice that is referenced in the summary invoice. | |
|  |  |  | | | | | | | | |  | | | | | Issuing date and time | | | | | | IssueDateTime | Attribute | [1, 1] | Date and time the invoice referenced by the summary invoice is created and issued at ENU. | |
|  |  | Delivery date or period | | | | | | | | | | | | | | | | | | | | SupplyDateOrPeriod | Element | [0, 1] | XML element representing supply date or period of supply, if it is different from the date when the invoice was issued.  Currently, the delivery date cannot be different from the invoice date, so this block will not be sent. | |
|  |  |  | | | | | | | | Start | | | | | | | | | | | | Start | Attribute | [1, 1] | Start day of the supply. | |
|  |  |  | | | | | | | | End | | | | | | | | | | | | End | Attribute | [1, 1] | End day of the supply. | |
|  |  | Payment methods | | | | | | | | | | | | | | | | | | | | PayMethods | Element | [1, 1] | XML element representing list of payment methods. | |
|  |  |  | Payment method | | | | | | | | | | | | | | | | | | | PayMethod | Element | [1, 10] | XML element representing one payment method. | |
|  |  |  |  | | | | | | | | | | Type | | | | | | | | | Type | Attribute | [1, 1] | Type of the payment method. | |
|  |  |  |  | | | | | | | | | | Amount | | | | | | | | | Amt | Attribute | [1, 1] | Amount paid by payment method in the currency in which the invoice was issued. It is mandatory if there are multiple payment methods. | |
|  |  |  |  | | | | | | | | | | Card issued by the company | | | | | | | | | CompCard | Attribute | [0, 1] | Number of card issued by the company if the payment method is company card. It is mandatory if the method / type of payment is stated: COMPANY. | |
|  |  |  |  | | | | | | | | | | Vouchers | | | | | | | | | Vouchers | Element | [0, 1] | XML element that contains list of voucher numbers if the payment method is SVOUCHER. It is mandatory if the method / type of payment is stated: SVOUCHER. | |
|  |  |  |  | | | | | | | | | |  | | | | Voucher | | | | | Voucher | Element | [1, 20] | XML element that contains one voucher number. | |
|  |  |  |  | | | | | | | | | |  | | | |  | | Number | | | Num | Attribute | [1, 1] | Voucher serial number. | |
|  |  | Currency | | | | | | | | | | | | | | | | | | | | Currency | Element | [0, 1] | XML element representing currency in which the amount on the invoice should be paid.  Only currency that can be sent is Euro (€). | |
|  |  |  | | | | | | Code | | | | | | | | | | | | | | Code | Attribute | [1, 1] | Currency code in which the amount on the invoice should be paid. | |
|  |  |  | | | | | | Exchange rate | | | | | | | | | | | | | | ExRate | Attribute | [1, 1] | Exchange rate applied to calculate the equivalent amount of foreign currency for the total amount expressed in €. Exchange rate express equivalent amount of € for 1 unit of foreign currency. | |
|  |  | Seller | | | | | | | | | | | | | | | | | | | | Seller | Element | [1, 1] | XML element representing seller’s data. | |
|  |  |  | | | | Identification type | | | | | | | | | | | | | | | | IDType | Attribute | [1, 1] | Seller’s identification number type. | |
|  |  |  | | | | Identification number | | | | | | | | | | | | | | | | IDNum | Attribute | [1, 1] | Seller’s identification number. | |
|  |  |  | | | | Name | | | | | | | | | | | | | | | | Name | Attribute | [1, 1] | Seller’s name. | |
|  |  |  | | | | Addressa | | | | | | | | | | | | | | | | Address | Attribute | [0, 1] | Seller’s address. | |
|  |  |  | | | | Town | | | | | | | | | | | | | | | | Town | Attribute | [0, 1] | Seller’s town. | |
|  |  |  | | | | Country | | | | | | | | | | | | | | | | Country | Attribute | [0, 1] | Seller’s country. | |
|  |  | Buyer | | | | | | | | | | | | | | | | | | | | Buyer | Element | [0, 1] | XML element representing buyer’s data. | |
|  |  |  | | | | | Identification type | | | | | | | | | | | | | | | IDType | Attribute | [1, 1] | Buyer’s identification number type. | |
|  |  |  | | | | | Identification number | | | | | | | | | | | | | | | IDNum | Attribute | [1, 1] | Buyer’s identification number. | |
|  |  |  | | | | | Name | | | | | | | | | | | | | | | Name | Attribute | [1, 1] | Buyer’s name. | |
|  |  |  | | | | | Address | | | | | | | | | | | | | | | Address | Attribute | [0, 1] | Buyer’s address. | |
|  |  |  | | | | | Town | | | | | | | | | | | | | | | Town | Attribute | [0, 1] | Buyer’s town. | |
|  |  |  | | | | | Country | | | | | | | | | | | | | | | Country | Attribute | [0, 1] | Buyer’s country. | |
|  |  | Items | | | | | | | | | | | | | | | | | | | | Items | Element | [1, 1] | XML element representing list of invoice items. | |
|  |  |  | | | | | | I (item) | | | | | | | | | | | | | | I (Item) | Element | [1, 1000] | XML element representing one item. | |
|  |  |  | | | | | |  | | | | | | | N (name) | | | | | | | N (Name) | Attribute | [1, 1] | Name of the item (goods or services). | |
|  |  |  | | | | | |  | | | | | | | C (code) | | | | | | | C (Code) | Attribute | [0, 1] | Code of the item from the barcode or similar representation. | |
|  |  |  | | | | | |  | | | | | | | IN (investment) | | | | | | | IN (Is Investment) | Attribute | [0, 1] | If the item is investment for the buyer. |
|  |  |  | | | | | |  | | | | | | | UM (unit of measure) | | | | | | | U (Unit of measure) | Attribute | [1, 1] | What is the item’s unit of measure (piece, weight measure, length measure, etc.) |
|  |  |  | | | | | |  | | | | | | | VS (vouchers sold) | | | | | | | VS (Voucher sold) | Element | [0, 1] | XML element representing vouchers sold | |
|  |  |  | | | | | |  | | | | | | |  | | | VD (voucher data) | | | | VD (Voucher data) | Element | [1, 1] | XML element representing data of vouchers sold |
|  |  |  | | | | | |  | | | | | | |  | | |  | | D (voucher validity date) | | D (Date) | Attribute | [1, 1] | Expiration date of the voucher. |
|  |  |  | | | | | |  | | | | | | |  | | |  | | N (voucher nominal value) | | N (Nominal value) | Attribute | [1, 1] | Nominal voucher value. |
|  |  |  | | | | | |  | | | | | | |  | | | VN (voucher numbers list) | | | | VN (Voucher sold numbers) | Element | [1, 1] | XML element representing serial numbers of voucher sold. |
|  |  |  | | | | | |  | | | | | | |  | | |  | | V (voucher number) | | V (Voucher number) | Element | [1, 1000] | XML element representing single voucher. |
|  |  |  | | | | | |  | | | | | | |  | | |  | |  | Num (voucher serial number) | Num (Number) | Attribute | [1, 1] | Voucher serial number. |
|  |  |  | | | | | |  | | | | | | | Q (quantity) | | | | | | | Q (Quantity) | Attribute | [1, 1] | Amount or number (quantity) of items. | |
|  |  |  | | | | | |  | | | | | | | UPB (unique price without VAT) | | | | | | | UPB (Unique price without VAT) | Attribute | [1, 1] | Unit price before VAT is applied. | |
|  |  |  | | | | | |  | | | | | | | UPA (unique price with VAT) | | | | | | | UPA (Unique price with VAT) | Attribute | [1, 1] | Unit price after VAT is applied. | |
|  |  |  | | | | | |  | | | | | | | R (rebate) | | | | | | | R (Rebate) | Attribute | [0, 1] | Percentage of the rebate. | |
|  |  |  | | | | | |  | | | | | | | RR (rebate reducing base price) | | | | | | | RR (Rebate reducing base price) | Attribute | [0, 1] | Is rebate reducing tax base amount or not | |
|  |  |  | | | | | |  | | | | | | | CP (price before VAT) | | | | | | | PB (Price before VAT) | Attribute | [1, 1] | Total price of goods and services before the tax. | |
|  |  |  | | | | | |  | | | | | | | VR (VAT rate) | | | | | | | VR (VAT rate) | Attribute | [0, 1] | Rate of VAT.  Mandatory if issuer is in PDV system. | |
|  |  |  | | | | | |  | | | | | | | EX (exempt from VAT) | | | | | | | EX (Exempt from VAT) | Attribute | [0, 1] | Exempt from VAT. | |
|  |  |  | | | | | |  | | | | | | | VA (VAT amount) | | | | | | | VA (VAT amount) | Attribute | [0, 1] | Amount of VAT for goods and services.  Mandatory if the issuer is in the VAT system. Mandatory if there is a self-charging device (vending machine) and the issuer is in the VAT system. Mandatory if reverse charging applies. | |
|  |  |  | | | | | |  | | | | | | | PA (Price after applying VAT) | | | | | | | PA (Price after applying VAT) | Attribute | [1, 1] | Total price of goods after the tax and applying discounts. | |
|  |  | Same tax items | | | | | | | | | | | | | | | | | | | | SameTaxes | Element | [0, 1] | XML element representing list of the aggregated items that go under same tax rate/exemption. There is only one XML element for each VAT rate. Mandatory if the issuer is in the VAT system. | |
|  |  |  | | | Same tax | | | | | | | | | | | | | | | | | SameTax | Element | [1, 20] | XML element representing one same tax item. | |
|  |  |  | | |  | | | | | | | | | Number of items | | | | | | | | NumOfItems | Attribute | [1, 1] | Number of items. | |
|  |  |  | | |  | | | | | | | | | Price before VAT | | | | | | | | PriceBefVAT | Attribute | [1, 1] | Price before VAT. | |
|  |  |  | | |  | | | | | | | | | VAT rate | | | | | | | | VATRate | Attribute | [1, 1] | VAT rate. | |
|  |  |  | | |  | | | | | | | | | Exempt from VAT | | | | | | | | ExemptFromVAT | Attribute | [1, 1] | Exempt from VAT. | |
|  |  |  | | |  | | | | | | | | | VAT amount | | | | | | | | VATAmt | Attribute | [1, 1] | VAT amount. | |
|  |  | Fees | | | | | | | | | | | | | | | | | | | | Fees | Element | [0, 1] | XML element representing list of fees. | |
|  |  |  | | Fee | | | | | | | | | | | | | | | | | | Fee | Element | [1, 20] | XML element representing one fee. | |
|  |  |  | |  | | | | | | | | Type | | | | | | | | | | Type | Attribute | [1, 1] | Type of the fee. | |
|  |  |  | |  | | | | | | | | Amount | | | | | | | | | | Amt | Attribute | [1, 1] | Amount of the fee. | |
|  |  | Bad debt | | | | | | | | | | | | | | | | | | | | BadDebtInv | Element | [0, 1] | XML element groups data for an original invoice that will be declared bad debt invoice, as uncollectible. |
|  |  |  | | | | | | IKOF reference | | | | | | | | | | | | | | IICRef | Attribute | [1, 1] | IKOF reference on the original invoice. |
|  |  |  | | | | | | Issuing date and time | | | | | | | | | | | | | | IssueDateTime | Attribute | [1, 1] | Date and time the original invoice is created and issued at ENU. |
|  | Signature | | | | | | | | | | | | | | | | | | | | | Signature | Element | [1, 1] | XML element representing signature for the invoice. | |

Table 43

#### Header

This is the XML element representing the header of the request data message.

#### Header: Universal unique identifier

Element generated by the ENU. It uniquely identifies the request message sent from ENU to CIS. UUID should be constructed according to the RFC4122 version 4.

|  |  |
| --- | --- |
| Data type | String |
| Length | 36 chars |
| Pattern | [0-9a-FA-F]{8}–[0-9a-FA-F]{4}–[1-5][0-9a-FA-F]{3}–[89abAB][0-9a-FA-F]{3}–[0-9a-FA-F]{12} |
| Example | 58e0a7d7-eebc-41d8-9669-0800200c9a66  58E0A7D7-EEBC-41D8-9669-0800200C9A66 |

Table 44

#### Header: Sending date and time

|  |  |
| --- | --- |
| Data type | Date and time |
| Pattern | [0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}([+-][0-9]{2}:[0-9]{2}|Z) |
| Example | 2019-01-24T22:00:58+01:00  2019-01-24T22:00:58-01:00 |

Table 45

#### Header: Subsequent delivery type

Element that says what is the type of subsequent delivery. For example, when the issuer sent the message to the CIS, but he did not receive the answer, he can send another message as before, but only with a flag with “Subsequent delivery” note. This flag indicates to CIS that the issuer sent another message.

|  |  |
| --- | --- |
| Data type | String |
| Value | ENUMERATION |
| Example | NOINTERNET |

Table 46

Table below shows the list of available values for Subsequent delivery type attribute.

|  |  |
| --- | --- |
| Value | Description |
| NOINTERNET | If ENU works in area without internet |
| BOUNDBOOK | ENU is not working and no message can be created |
| SERVICE | Fiscalization service problem |
| TECHNICALERROR | Technical error |
| BUSINESSNEED | Subsequent sending conditioned by the way of doing business |

Table 47

#### Invoice

XML element representing a single invoice.

#### Invoice: Type

This attribute represents the type of the invoice in question.

|  |  |
| --- | --- |
| Data type | String |
| Value | ENUMERATION |
| Example | CASH |

Table 48

The table below shows the list of values for Type attribute

|  |  |
| --- | --- |
| Value | Description |
| CASH | Cash invoice |
| NONCASH | Non-cash invoice |

Table 49

#### Invoice: Invoice type issued by the customer

This element shows the type of the invoice that is issued by the customer himself.

|  |  |
| --- | --- |
| Data type | String |
| Value | ENUMERATION |
| Example | ABROAD |

Table 50

The table below shows the list of values for Invoice type issued by the customer.

|  |  |
| --- | --- |
| Value | Description |
| AGREEMENT | Previous agreement by the sides |
| DOMESTIC | Buying from local |
| ABROAD | Buying services abroad. |
| OTHER | Other |

Table 51

#### Invoice: Simplified invoice

Is the invoice simplified?

|  |  |
| --- | --- |
| Data type | Boolean |
| Value | true OR false |
| Example | true |

Table 52

#### Invoice: Issuing date and time

Date and time when ENU creates and issues the invoice.

|  |  |
| --- | --- |
| Data type | Date and time |
| Pattern | [0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}([+-][0-9]{2}:[0-9]{2}|Z) |
| Example | 2019-01-24T22:00:58+01:00  2019-01-24T22:00:58-01:00 |

Table 53

#### Invoice: Invoice number

Invoice number consisted of business unit code, invoice ordinal number, year of issuing and ENU code. Ordinal number is a subsequent number for each new invoice and is reset on start of each year.

|  |  |
| --- | --- |
| Data type | String |
| Pattern | [a-z]{2}[0-9]{3}[a-z]{2}[0-9]{3}\/[1-9]{1}[0-9]{0,14}\/[0-9]{4}\/[a-z]{2}[0-9]{3}[a-z]{2}[0-9]{3} |
| Example | pp123pp123/9934/2019/ab123ab123 |

Table 54

#### Invoice: Ordinal number

Ordinal number of the invoice. Each new invoice gets the next number, counting starts from zero on start of a year.

|  |  |
| --- | --- |
| Data type | Integer |
| Limit | Positive |
| Example | 2 |

Table 55

#### Invoice: ENU code

Code of the ENU that issued the invoice.

|  |  |
| --- | --- |
| Data type | String |
| Pattern | [a-z]{2}[0-9]{3}[a-Z]{2}[0-9]{3} |
| Example | ab123ab123 |

Table 56

#### Invoice: Issuer is in VAT

Is issuer in VAT system?

|  |  |
| --- | --- |
| Data type | Boolean |
| Value | true OR false |
| Example | true |

Table 57

#### Invoice: Amount without VAT

Amount of the VAT free invoice, that may depend on various reasons.

|  |  |
| --- | --- |
| Data type | Decimal |
| Pattern | -?([1-9][0-9]\*|0)\[0-9]{2}|0 |
| Example | 17.24 |

Table 58

#### Invoice: Margin amount

Total amount regarding special margin process on the invoice written in decimal form (taxable amount). Margin for used goods, works of art, antiquities or collectible items.

|  |  |
| --- | --- |
| Data type | Decimal |
| Pattern | -?([1-9][0-9]\*|0)\.[0-9]{2}|0 |
| Example | 23.10 |

Table 59

#### Invoice: Amount of export of goods

Total price of the exported goods. There is no VAT on the invoice.

|  |  |
| --- | --- |
| Data type | Decimal |
| Pattern | -?([1-9][0-9]\*|0)\.[0-9]{2}|0 |
| Example | 246.00 |

Table 60

#### Invoice: Total invoice price without VAT

Total invoice price without VAT.

|  |  |
| --- | --- |
| Data type | Decimal |
| Pattern | -?([1-9][0-9]\*|0)\.[0-9]{2}|0 |
| Example | 212.12 |

Table 61

#### Invoice: Total VAT amount

Total VAT amount that is needed to be paid.

|  |  |
| --- | --- |
| Data type | Decimal |
| Pattern | -?([1-9][0-9]\*|0)\.[0-9]{2}|0 |
| Example | 242.23 |

Table 62

#### Invoice: Total price

Total price of the invoice that is needed to be paid.

|  |  |
| --- | --- |
| Data type | Decimal |
| Pattern | -?([1-9][0-9]\*|0)\.[0-9]{2}|0 |
| Example | 212.12 |

Table 63

#### Invoice: Operator code

Code of the operator using the ENU.

|  |  |
| --- | --- |
| Data type | String |
| Length | 10 chars |
| Pattern | [a-z]{2}[0-9]{3}[a-Z]{2}[0-9]{3} |
| Example | ab123ab123 |

Table 64

#### Invoice: Business unit code

Code of the business unit where the invoice is issued.

|  |  |
| --- | --- |
| Data type | String |
| Length | 10 chars |
| Pattern | [a-z]{2}[0-9]{3}[a-Z]{2}[0-9]{3} |
| Example | ab123ab123 |

Table 65

#### Invoice: Software code

Code of the software used for the invoice issuing.

|  |  |
| --- | --- |
| Data type | String |
| Length | 10 chars |
| Pattern | [a-z]{2}[0-9]{3}[a-Z]{2}[0-9]{3} |
| Example | ab123ab123 |

Table 66

#### Invoice: IKOF

Invoice identification code generated by the ENU. This is the unique code and each fiscalized invoice contains one. Details on IKOF can be found in chapter **Pogreška! Izvor reference nije pronađen.**.

|  |  |
| --- | --- |
| Data type | String |
| Length | 32 |
| Pattern | [0-9a-FA-F]{32} |
| Example | C701FB4839E7D2C3D8DBC81BBAC06164  c701fb4839e7d2c3d8dbc81bbac06164 |

Table 67

#### Invoice: IKOF signature

Connected parameters of the IKOF. Details can be found in chapter **Pogreška! Izvor reference nije pronađen.**.

|  |  |
| --- | --- |
| Data type | String |
| Length | 512 |
| Pattern | [0-9a-FA-F]{512} |
| Example ||

Table 68

#### Invoice: Corrective invoice

XML element that groups the data for the invoice that is going to be corrected.

#### Corrective invoice: Reference on IKOF

Reference on the original invoice. Inserted only if this is corrective invoice.

|  |  |
| --- | --- |
| Data type | String |
| Length | 32 |
| Pattern | [0-9a-FA-F]{32} |
| Example | C701FB4839E7D2C3D8DBC81BBAC06164  c701fb4839e7d2c3d8dbc81bbac06164 |

Table 69

#### Corrective invoice: Issuing date and time

Date and time of issuing of the original invoice that is being corrected.

|  |  |
| --- | --- |
| Data type | Date and time |
| Pattern | [0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}([+-][0-9]{2}:[0-9]{2}|Z) |
| Example | 2019-01-24T22:00:58+01:00  2019-01-24T22:00:58-01:00 |

Table 70

#### Corrective invoice: Type

Type of the corrective invoice.

|  |  |
| --- | --- |
| Data type | String |
| Value | ENUMERATION |
| Example | CORRECTIVE |

Table 71

The table below shows the list of values for Type.

|  |  |
| --- | --- |
| Value | Description |
| CORRECTIVE | Corrective invoice |
| DEBIT | Debit note |
| CREDIT | Credit note |

Table 72

#### Invoice: Reverse charge

Buyer is obligated to pay the VAT by himself instead. Note: Reverse charge is not currently subject to fiscalization, so it is necessary to set the value “FALSE” in this field.

|  |  |
| --- | --- |
| Data type | Boolean |
| Value | true OR false |
| Example | false |

Table 73

#### Invoice: Bad debt

If the invoice is flagged as the one that cannot be paid, it is shown as “bad debt”.

|  |  |
| --- | --- |
| Data type | Boolean |
| Value | true OR false |
| Example | true |

Table 74

#### Invoice: Payment deadline

Last date to pay the invoice.

|  |  |
| --- | --- |
| Data type | String |
| Length | 10 chars |
| Pattern | [0-9]{4}–[0-9]{2}–[0-9]{2} |
| Example | 2019-01-24 |

Table 75

#### Summary invoice with a list of IKOFs it is referencing

XML element that contains a list of unique invoice identification codes (IKOFs) to which this summary invoice refers, e.g. individual invoices included in the summary invoice, and whose method of payment was COMPANY or ORDER.

#### Summary invoice with a list of IKOFs it is referencing: Referent unique IKOF

Summary invoice with a list of IKOF references

#### Referent unique IKOF: Single invoice IKOF

A single IKOF that is referenced on the list of IKOFs of an summary invoice.

|  |  |
| --- | --- |
| Data type | String |
| Length | 32 |
| Pattern | [0-9a-FA-F]{32} |
| Example | C701FB4839E7D2C3D8DBC81BBAC06164  c701fb4839e7d2c3d8dbc81bbac06164 |

Table 76

#### Referent unique IKOF: Date and time of issuing the invoice

Date and time when the invoice of the referenced IKOF was issued.

|  |  |
| --- | --- |
| Data type | Date and time |
| Pattern | [0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}([+-][0-9]{2}:[0-9]{2}|Z) |
| Example | 2019-01-24T22:00:58+01:00  2019-01-24T22:00:58-01:00 |

Table 77

#### Invoice: Delivery date or period

Element that represents the date or period of delivery if it is different than the issuing date.

#### Delivery date or period: Start

Date of delivery start. If delivery starts and ends on the same date, Start is the same as End. If there is period of multiple days, this is first day od the period. Period cannot last in multiple months.

|  |  |
| --- | --- |
| Data type | String |
| Length | 10 chars |
| Pattern | [0-9]{4}–[0-9]{2}–[0-9]{2} |
| Example | 2019-01-24 |

Table 78

#### Delivery date or period: End

Date of delivery ending.

|  |  |
| --- | --- |
| Data type | String |
| Length | 10 chars |
| Pattern | [0-9]{4}–[0-9]{2}–[0-9]{2} |
| Example | 2019-01-24 |

Table 79

#### Invoice: Payment methods

XML element representing the list of invoice payment method.

#### Payment methods: Payment method

XML element representing a single payment method.

#### Payment method: Type

|  |  |
| --- | --- |
| Data type | String |
| Value | ENUMERATION |
| Example | BANKNOTE |

Table 80

The table below shows the list of values for payment method type.

|  |  |  |
| --- | --- | --- |
| **Value** | **Description** | **Allowed invoice type** |
| BANKNOTE | Notes and coins | CASH |
| CARD | Credit or debit card of the bank issued to a natural person | CASH |
| BUSINESSCARD | Credit or debit card of the bank issued to taxpayer | NONCASH |
| SVOUCHER | Onetime voucher | NONCASH |
| COMPANY | Cards issued by the company, gift cards and similar prepaid cards | NONCASH |
| ORDER | Invoice to be paid in summary invoice | CASH and NONCASH |
| ADVANCE | Advance payment | NONCASH |
| ACCOUNT | Transaction account | NONCASH |
| FACTORING | Factoring | NONCASH |
| OTHER | Other non-cash payment types | NONCASH |
| OTHER-CASH | Other cash payment types | CASH |

Table 81

#### Payment method: Amount

Total amount payed with a single payment method.

|  |  |
| --- | --- |
| Data type | Decimal |
| Pattern | -?([1-9][0-9]\*|0)\.[0-9]{2}|0 |
| Example | 212.12 |

Table 82

#### Payment method: Company card

Company card number if the payment method is COMPANY.

|  |  |
| --- | --- |
| Data type | String |
| Length | 50 chars |
| Example | 1257896 |

Table 83

#### Payment method: Voucher

Element containing number of vouchers if the payment method is SVOUCHER. These can be multiple vouchers used for the same invoice (i.e. for the same delivery, but one voucher cannot be shared for 2 or more deliveries).

#### Vouchers: Voucher

Element containing one voucher number.

#### Voucher: Voucher number

Voucher number.

|  |  |
| --- | --- |
| Data type | String |
| Length | 17-25 chars |
| Pattern | [1-9][0-9]{0,7}–[0-9]{4}–[0-9]{8} |
| Example | 2-2020-12345678  34564-2020-12345678 |

Table 84

#### Invoice: Currency

Element showing the invoice currency

#### Currency: Code

Currency cod that uses ISO 4217 format.

|  |  |
| --- | --- |
| Data type | String |
| Value | Shown in table below |
| Example | EUR |

Table 85

The table below shows the list of values for currency codes. Other values can be found in chapter 7.

|  |  |
| --- | --- |
| Value | Description |
| EUR | Euro |
| ALL | Albanian lek |
| GRD | Greek drahma |
| MKD | North Macedonian denar |
| TRY | Turkish lira |
| BGN | Bulgarian lev |
| BAM | Bosnian Herzegovina convertibile mark |
| HRK | Croatian kuna |
| … | .. |

Table 86

#### Currency: Exchange rate

Exchange rate if foreign currency is in use.

|  |  |
| --- | --- |
| Data type | Double |
| Limit | Must be positive number. |
| Example | 3.500  0.375 |

Table 87

#### Invoice: Paragon block number

If the invoice was created on a paragon block, its number is inserted.

|  |  |
| --- | --- |
| Data type | String |
| Limit | 20 chars |
| Example | 5-2020 |

Table 88

#### Invoice: Tax period

Tax period that invoice belongs.

|  |  |
| --- | --- |
| Data type | String |
| Pattern | ((0[1-9])|(1[0-2]))\/(\d{4}) |
| Example | 08/2020 |

Tabela 89

#### Invoice: Issuer

XML element representing the invoice issuer.

#### Issuer: Identification code

Issuer ID code

|  |  |
| --- | --- |
| Data type | String |
| Length | 20 chars |
| Example | PIB/TIN: 72001008  Social security number: 123-45-6789 |

Table 90

#### Identification code: Type

Identification code type.

|  |  |
| --- | --- |
| Data type | String |
| Value | ENUMERATION |
| Example | TIN |

Table 91

Table below shows the list of available values for identification code type attribute.

|  |  |
| --- | --- |
| Value | Description |
| TIN | Personal tax number |
| ID | Personal identification number |
| PASS | Passport number |
| VAT | VAT number |
| TAX | Tax number |
| SOC | Social security number |

Table 92

#### Issuer: Name

Issuer name.

|  |  |
| --- | --- |
| Data type | String |
| Length | 100 chars |
| Example | Petar Petrović |

Table 93

#### Issuer: Address

Issuer address.

|  |  |
| --- | --- |
| Data type | String |
| Length | 200 chars |
| Example | Tivatska ulica 27 |

Table 94

#### Issuer: Town

Issuer town.

|  |  |
| --- | --- |
| Data type | String |
| Length | 100 chars |
| Example | Berane |

Table 95

#### Issuer: Country

Issuer country shown as ISO 3166-1 Alfa-3 code.

|  |  |
| --- | --- |
| Data type | String |
| Value | ENUMERATION |
| Example | MNE |

Table 96

Values for some countries are shown in table below. Rest is shown in chapter 7.

|  |  |
| --- | --- |
| Value | Description |
| MNE | Montenegro |
| ALB | Albania |
| GRC | Greece |
| MKD | North Macedonia |
| RKS | Kosovo |
| ITA | Italy |
| … | … |

Table 97

#### Invoice: Buyer

XXML representing the buyer whom the invoice is issued to.

#### Buyer: Identification code

Buyer identification code.

|  |  |
| --- | --- |
| Data type | String |
| Length | 20 chars |
| Example | PIB/TIN: 72001008  Social security number: 123-45-6789 |

Table 98

#### Identification code: Type

Identification code type.

|  |  |
| --- | --- |
| Data type | String |
| Value | ENUMERATION |
| Example | TIN |

Table 99

Table below shows the list of available values for identification code type attribute.

|  |  |
| --- | --- |
| Value | Description |
| TIN | Personal tax number |
| ID | Personal identification number |
| PASS | Passport number |
| VAT | VAT number |
| TAX | Tax number |
| SOC | Social security number |

Table 100

#### Buyer: Name

Buyer name

|  |  |
| --- | --- |
| Data type | String |
| Length | 100 chars |
| Example | Ivan Ivanović |

Table 101

#### Buyer: Address

Buyer address

|  |  |
| --- | --- |
| Data type | String |
| Length | 200 chars |
| Example | Barska ulica 2 |

#### Buyer: Town

Buyer town

|  |  |
| --- | --- |
| Data type | String |
| Length | 100 chars |
| Example | Podgorica |

Table 102

#### Buyer: Country

Buyer country shown in ISO 3166-1 Alfa-3 code.

|  |  |
| --- | --- |
| Data type | String |
| Value | ENUMERATION |
| Example | MNE |

Table 103

Values for some countries are shown in table below. Rest is shown in chapter 7.

|  |  |
| --- | --- |
| Value | Description |
| MNE | Montenegro |
| ALB | Albania |
| GRC | Greece |
| MKD | North Macedonia |
| RKS | Kosovo |
| ITA | Italy |

Table 104

#### Invoice: Items

XML element representing a list of items on the invoice (goods or services). Items that are the same should be grouped as one item (one XML element called "item") with the appropriate quantity (sum of the same items).

When calculating on items, all amounts must be expressed with the same number of decimals. For example, if 4 decimals are displayed in one place and 4 decimals must be displayed in all other places on the items. Only after grouping and summation for the same tax and for the total amount are the amounts rounded to 2 decimal places and a tolerance of 0.01 € is taken during this rounding.

#### Invoice: I (items)

XML element representing a single item on the invoice.

#### Item: N (name)

Name of the item.

|  |  |
| --- | --- |
| Data type | String |
| Length | 50 chars |
| Example | vino 1.5L |

Table 105

#### Item: C (code)

Item unique code (barcode).

|  |  |
| --- | --- |
| Data type | String |
| Length | 50 chars |
| Example | 978020137962 |

Table 106

#### Item: IN (investment)

Flag that the item is an investment.

|  |  |
| --- | --- |
| Data type | Boolean |
| Value | true OR false |
| Example | true |

Table 107

#### Item: VS (vouchers sold)

XML element containing information on vouchers used on this invoice.

#### Vouchers sold: VD (voucher data)

XML element containing data of sold vouchers.

#### Voucher Data: D (voucher validity date)

Date of voucher validity.

|  |  |
| --- | --- |
| Data type | String |
| Length | 10 chars |
| Pattern | [0-9]{4}–[0-9]{2}–[0-9]{2} |
| Example | 2019-01-24 |

Table 108

#### Voucher data: N (voucher nominal value)

Nominal voucher value.

|  |  |
| --- | --- |
| Data type | Decimal |
| Pattern | -?([1-9][0-9]\*|0)\.[0-9]{2,4}|0 |
| Example | 3.5056 |

Table 109

#### Vouchers sold: VN (voucher numbers list)

XML element containing a list of vouchers’ numbers.

#### Voucher numbers list: V (voucher number)

XML element containing a single voucher number

#### Voucher number: Num (voucher serial number)

Voucher serial number

|  |  |
| --- | --- |
| Data type | String |
| Length | 17-25 chars |
| Pattern | [1-9][0-9]{0,7}–[0-9]{4}–[0-9]{8} |
| Example | 2-2020-12345678  34564-2020-12345678 |

Table 110

#### Item: UM (unit of measure)

Unit of measure for specific item – kilo, piece…

|  |  |
| --- | --- |
| Data type | String |
| Length | 50 chars |
| Example | Kg |

Table 111

#### Item: Q (quantity)

Quantity or number of items.

|  |  |
| --- | --- |
| Data type | Double |
| Pattern | -?([1-9][0-9]\*|0)(\.[0-9]{1,3})? |
| Example | 3.500  -0.375 |

Table 112

#### Item: UPB (unique price without VAT)

Price of a single item before VAT is applied.

|  |  |
| --- | --- |
| Data type | Decimal |
| Pattern | ([1-9][0-9]\*|0)\.[0-9]{2,4}|0 |
| Example | 3.5005 |

Table 113

#### Item: UPA (unique price with VAT)

Price of a single item after VAT is applied.

|  |  |
| --- | --- |
| Data type | Decimal |
| Pattern | ([1-9][0-9]\*|0)\.[0-9]{2,4}|0 |
| Example | 3.8501 |

Table 114

#### Item: R (rebate)

Rebate percentage.

|  |  |
| --- | --- |
| Data type | Decimal |
| Pattern | ([1-9][0-9]\*|0)\.[0-9]{2,4}|0 |
| Example | 33.1767 |

Table 115

#### Item: RR (rebate reducing base price)

Is rebate reducing the base price or not.

|  |  |
| --- | --- |
| Data type | Boolean |
| Values | true OR false |
| Example | true |

Table 116

#### Item: PB (price before VAT)

Price of all items of this group before VAT is applied.

|  |  |
| --- | --- |
| Data type | Decimal |
| Pattern | -?([1-9][0-9]\*|0)\.[0-9]{2,4}|0 |
| Example | 134.3401 |

Table 117

#### Item: VR (VAT rate)

VAT rate written as percentage.

|  |  |
| --- | --- |
| Data type | Decimal |
| Pattern | -?([1-9][0-9]\*|0)\.[0-9]{2,4}|0 |
| Example | 3.5001 |

Table 118

#### Item: EX (exempt/released from VAT)

Exemption/release from paying the VAT.

|  |  |
| --- | --- |
| Data type | String |
| Value | ENUMERATION |
| Example | VAT\_CL26 |

Table 119

Values exempt/released from VAT are listed in table below.

|  |  |
| --- | --- |
| Values | Description |
| VAT\_CL17 | Place of service turnover |
| VAT\_CL20 | Tax base and correction of the tax base |
| VAT\_CL26 | Disengagement from public interest |
| VAT\_CL27 | Other disengagements |
| VAT\_CL28 | Disengagement when importing goods |
| VAT\_CL29 | Disengagement when importing goods temporarily |
| VAT\_CL30 | Special disengagements |

Table 120

#### Item: VA (VAT amount)

VAT amount for this group of items.

|  |  |
| --- | --- |
| Data type | Decimal |
| Pattern | -?([1-9][0-9]\*|0)\.[0-9]{2,4}|0 |
| Example | 3.5001 |

Table 121

#### Item: PA (price after applying VAT)

Price with VAT applied for this group of items.

|  |  |
| --- | --- |
| Data type | Decimal |
| Pattern | -?([1-9][0-9]\*|0)\.[0-9]{2,4}|0 |
| Example | 3.5099 |

Table 122

#### Invoice: Same taxes

XML element represents a list of items in the invoice (goods or services) to which the same VAT rate is applied and / or are exempt from VAT. **All items to which the same rate applies / exempt from VAT are grouped together.** There is only one XML element for each VAT rate.

#### Same taxes: Same tax

XML element representing the items that have the same VAT rate.

#### Same tax: Number of items

Number of items with the same VAT rate.

|  |  |
| --- | --- |
| Data type | Integer |
| Limit | positive |
| Example | 2 |

Table 123

#### Same tax: Price before VAT

Price of items of same tax rate before VAT is applied.

|  |  |
| --- | --- |
| Data type | Decimal |
| Pattern | -?([1-9][0-9]\*|0)\.[0-9]{2}|0 |
| Example | 12.20 |

Table 124

#### Same tax: VAT rate

VAT rate for items of same tax rate.

|  |  |
| --- | --- |
| Data type | Decimal |
| Pattern | -?([1-9][0-9]\*|0)\.[0-9]{2}|0 |
| Example | 10.00 |

Table 125

#### Same tax: Exempt/released from VAT

Exempt/released from VAT for items of same tax rate.

|  |  |
| --- | --- |
| Data type | String |
| Value | ENUMERATION |
| Example | VAT\_CL26 |

Table 126

Values for tax exemption/release are shown in table below.

|  |  |
| --- | --- |
| Values | Description |
| VAT\_CL17 | Place of service turnover |
| VAT\_CL20 | Tax base and correction of the tax base |
| VAT\_CL26 | Disengagement from public interest |
| VAT\_CL27 | Other disengagements |
| VAT\_CL28 | Disengagement when importing goods |
| VAT\_CL29 | Disengagement when importing goods temporarily |
| VAT\_CL30 | Special disengagements |

Table 127

#### Same tax: VAT amount

VAT amount for items of same tax rate.

|  |  |
| --- | --- |
| Data type | Decimal |
| Pattern | -?([1-9][0-9]\*|0)\.[0-9]{2}|0 |
| Example | 246.00 |

Table 128

#### Invoice: Fees

XML element representing the list of fees paid on the invoice.

#### Fees: Fee

XML element representing a single fee.

#### Fee: Type

Fee type.

|  |  |
| --- | --- |
| Data type | String |
| Value | ENUMERATION |
| Example | PACK |

Table 129

Values for fee types are shown in table below.

|  |  |
| --- | --- |
| Value | Description |
| PACK | Packing fee |
| BOTTLE | Return glass bottles fee |
| COMMISION | Money exchange office commission |
| OTHER | Other fees |

Table 130

#### Fee: Amount

Decimal amount of the fee.

|  |  |
| --- | --- |
| Data type | Decimal |
| Pattern | -?([1-9][0-9]\*|0)\.[0-9]{2}|0 |
| Example | 12.20 |

Table 131

#### Invoice: Bad debt

XML element that groups the data for the invoice that is going to be flagged as not payable.

#### Bad debt: IKOF reference

Referencing the IKOF of the original invoice.

|  |  |
| --- | --- |
| Data type | String |
| Length | 32 |
| Pattern | [0-9a-FA-F]{32} |
| Example | C701FB4839E7D2C3D8DBC81BBAC06164  c701fb4839e7d2c3d8dbc81bbac06164 |

Table 132

#### Bad debt: Issuing date and time

Date and time of the original invoice issuing on the ENU.

|  |  |
| --- | --- |
| Data type | Date and time |
| Pattern | [0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}([+-][0-9]{2}:[0-9]{2}|Z) |
| Example | 2019-01-24T22:00:58+01:00  2019-01-24T22:00:58-01:00 |

Table 133

#### Signature

XML element stores enveloped digital signature described in the chapter **Pogreška! Izvor reference nije pronađen.**.

### REGISTER INVOICE RESPONSE MESSAGE

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | | | **XML tag** | **Field type** | **Occurrence[min, max]** | **Description** |
| **Response message for register invoice** | | | RegisterInvoiceResponse | Element | [1, 1] | Root XML element representing register invoice response message. |
|  | Identifier | | ID | Attribute | [1, 1] | Attribute used for signature creation and verification. Fixed value “Request”. |
|  | Version | | Version | Attribute | [1, 1] | Attribute used to specify compliance with XSD schema. For this version fixed value is “3”. |
|  | Header | | Header | Element | [1, 1] | XML element representing header of the invoice containing data about the message (request) sent. |
|  |  | Universal unique identifier | UUID | Attribute | [1, 1] | UUID generated by CIS for each registered electronic response message. |
|  |  | Universal unique request identifier | RequestUUID | Attribute | [1, 1] | UUID of the request message. |
|  |  | Sending date and time | SendDateTime | Attribute | [1, 1] | Date and time of sending the message. |
|  | Fiscalization identifying code (JIKR) | | FIC | Element | [1, 1] | Verification code generated by the CIS. |
|  | Signature | | Signature | Element | [1, 1] | XML element with the signature |

Table 134

#### Header

Element representing the header of the request data message.

#### Header: Universal unique identifier

Element generated by the ENU. It uniquely identifies the request message sent from ENU to CIS. UUID should be constructed according to the RFC4122 version 4.

|  |  |
| --- | --- |
| Data type | String |
| Length | 36 chars |
| Pattern | [0-9a-FA-F]{8}–[0-9a-FA-F]{4}–[1-5][0-9a-FA-F]{3}–[89abAB][0-9a-FA-F]{3}–[0-9a-FA-F]{12} |
| Example | 58e0a7d7-eebc-41d8-9669-0800200c9a66  58E0A7D7-EEBC-41D8-9669-0800200C9A66 |

Table 135

#### Header: Universal unique request identifier

Element generated by the ENU and referenced by the CIS. It uniquely identifies the request message for which the response message was sent to the ENU. UUID should be constructed according to the RFC4122 version 4.

|  |  |
| --- | --- |
| Data type | String |
| Length | 36 chars |
| Pattern | [0-9a-FA-F]{8}–[0-9a-FA-F]{4}–[1-5][0-9a-FA-F]{3}–[89abAB][0-9a-FA-F]{3}–[0-9a-FA-F]{12} |
| Example | 58e0a7d7-eebc-41d8-9669-0800200c9a66  58E0A7D7-EEBC-41D8-9669-0800200C9A66 |

Table 136

#### Header: Sending date and time

Element representing date and time of sending the response to the ENU.

|  |  |
| --- | --- |
| Data type | Date and time |
| Pattern | [0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}([+-][0-9]{2}:[0-9]{2}|Z) |
| Example | 2019-01-24T22:00:58+01:00  2019-01-24T22: 00: 58-01: 00 |

Table 137

#### Fiscalization identification code

Element representing the unique identification code when issuing the invoice.

|  |  |
| --- | --- |
| Data type | String |
| Length | 36 chars |
| Pattern | [0-9a-FA-F]{8}–[0-9a-FA-F]{4}–[1-5][0-9a-FA-F]{3}–[89abAB][0-9a-FA-F]{3}–[0-9a-FA-F]{12} |
| Example | 58e0a7d7-eebc-41d8-9669-0800200c9a66  58E0A7D7-EEBC-41D8-9669-0800200C9A66 |

Table 138

#### Signature

XML element stores enveloped digital signature described in the chapter **Pogreška! Izvor reference nije pronađen.**.

### MANDATORY CONTROLS

Mandatory controls shall be performed on received invoices data messages in the CIS system in real time. When any of the critical controls return a failure, the invoice data message shall not be approved, and JIKR shall not be issued.

Upon identifying a critical error, CIS will return an error data message containing the error's numeric code and its text description (see chapter **Pogreška! Izvor reference nije pronađen.**). When errors which the system can interpret as a cyber-attack are identified, the system does not send any response to the client (the ENU).

The mandatory controls include all the controls from chapter 3.4 and the following:

|  |  |  |
| --- | --- | --- |
| **Control name** | **Control description** | **Error code** |
| PIB identification type | Identification type is not PIB (TIN). | 54 |
| Taxpayer does not exist | Taxpayer does not exist in Register of taxpayers. | 52 |
| Taxpayer status is not valid | Taxpayer is inactive in Register of taxpayers. | 55 |
| Issuer tax status is invalid | Issuer status is different than the real issuer tax status.. | 44 |
| VAT status on the invoice is invalid | Issuer is in VAT system, no VAT is shown on the invoice.  Issuer is in VAT system, but there is no invoice.  Issuer is not in VAT system, VAT is shown on the invoice, reverse charge is incorrect.  Issuer is not in VAT system, same taxes element is shown on the invoice.  Self-vending machine type is shown, but there is no invoice.  Reverse charge exists, but there is no invoice. | 11 |

Table 139

Except for mandatory controls that will be done in real time after receiving the invoice data in CIS and done fiscalization there will be additional checks. If there are problems during the additional checks, the errors will be shown to a taxpayer on the Self-care porta and the taxpayer will have to send the corrective invoice. There are two groups of controls:

* Amount controls
  + Total amount of all items under the same tax rate
  + Total VAT amount
  + Total amount without VAT
  + Total amount with VAT
  + Total amount with VAT for items
  + Tax base value
  + VAT control
  + Single price with VAT
* Business checks
  + Issuer status
  + Valid issuer PIB
  + Valid ENU code
  + Valid business unit code
  + Valid operator code
  + Valid software code
  + Corrective invoice reference
  + Initial deposit
  + Valid voucher
  + VAT rate exists
  + Invoice summary
  + Message UUID

### ERROR MESSAGE

Error message is defined in chapter **Pogreška! Izvor reference nije pronađen.**.

### XML EXAMPLE

#### XML request

<SOAP-ENV:Envelope xmlns:SOAP-ENV=**"http://schemas.xmlsoap.org/soap/envelope/"**>

<SOAP-ENV:Header/>

<SOAP-ENV:Body>

<RegisterInvoiceRequest xmlns=**" https://efi.tax.gov.me/fs/schema "** xmlns:ns2=**"http://www.w3.org/2000/09/xmldsig#"** Id=**"Request"** Version=**"1"**>

<Header SendDateTime=**"2019-12-05T14:30:13+01:00"** UUID=**"8d216f9a-55bb-445a-be32-30137f11b964"** />

<Invoice BusinUnitCode=**"ab123ab123"** IssueDateTime=**"2019-12-05T14:30:13+01:00"** IIC=**"4AD5A215BEAF85B0416235736A6DACAB"** IICSignaturenvNum=**"1/2019/cc123cc123 (KOJI FORMAT BROJA RAČUNA"** InvOrdNum=**"1"** IsIssuerInVAT=**"true"** IsReverseCharge=**"false"** IsSimplifiedInv=**"false"** OperatorCode=**"ab123ab123"** SoftCode=**"EXAMPLE KODA SOFTVERA"** TCRCode=**"KOD BLAGAJNE"** TotPrice=**"20.00"** TotPriceWoVAT=**"16.00"** TotVATAmt=**"4.00"** TypeOfInv=**"CASH"**>

<PayMethods>

<PayMethod Amt=**"20.00"** Type=**"BANKNOTE"**/>

</PayMethods>

<Seller Address=**"ADRESA PRODAVAOCA"** Country=**"MNE"** IDNum=**"ID BROJ PRODAVAOCA"** IDType=**"PIB"** Name=**"IME PREZIME PRODAVAOCA"** Town=**"GRAD PRODAVAOCA"**/>

<Items>

<I C=**"501234567890"** N=**"NAZIV PROIZVODA"** PA=**"20.00"** PB=**"16.00"** Q=**"1.0"** R=**"0"** RR=**"true"** U=**"piece"** UPB=**"16.00"** UPA="20.00" VA=**"4.00"** VR=**"25.00"**/>

</Items>

<SameTaxes>

<SameTax NumOfItems=**"1"** PriceBefVAT=**"16.00"** VATAmt=**"4.00"** VATRate=**"25.00"**/>

</SameTaxes>

</Invoice>

<Signature xmlns=**"http://www.w3.org/2000/09/xmldsig#"**>

<SignedInfo>

<CanonicalizationMethod Algorithm=**"http://www.w3.org/2001/10/xml-exc-c14n#"**/>

<SignatureMethod Algorithm=**"http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"**/>

<Reference URI=**"#Request"**>

<Transforms>

<Transform Algorithm=**"http://www.w3.org/2000/09/xmldsig#enveloped-signature"**/>

<Transform Algorithm=**"http://www.w3.org/2001/10/xml-exc-c14n#"**/>

</Transforms>

<DigestMethod Algorithm=**"http://www.w3.org/2001/04/xmlenc#sha256"**/>

<DigestValue>**ECWby0FKaNQ6..........m8BPjyk6g=**</DigestValue>

</Reference>

</SignedInfo>

<SignatureValue>**LHFUg3xNi..........tjpeYNw==**</SignatureValue>

<KeyInfo>

<X509Data>

<X509Certificate>**MIIFYDCC..........Pifz0UlKJAanmqN3**</X509Certificate>

</X509Data>

</KeyInfo>

</Signature>

</RegisterInvoiceRequest>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

#### XML response

<SOAP-ENV:Envelope xmlns:SOAP-ENV=**"http://schemas.xmlsoap.org/soap/envelope/"**>

<SOAP-ENV:Header/>

<SOAP-ENV:Body>

<ns2:RegisterInvoiceResponse Id=**"Response" Version="1"** xmlns:ns2=**" https://efi.tax.gov.me/fs/schema "** xmlns:ns3=**"http://www.w3.org/2000/09/xmldsig#"**>

<ns2:Header RequestUUID=**"8d216f9a-55bb-445a-be32-30137f11b964"** SendDateTime=**"2019-11-27T09:41:12+01:00"** UUID=**"f8bcb5ae-59fb-41ac-9011-f4db86bbce26"**/>

<ns2:FIC>**a592e7ec-9517-4f02-8d54-ac965f679a8c**</ns2:FIC>

<Signature xmlns=**"http://www.w3.org/2000/09/xmldsig#"**>

<SignedInfo>

<CanonicalizationMethod Algorithm=**"http://www.w3.org/2001/10/xml-exc-c14n#"**/>

<SignatureMethod Algorithm=**"http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"**/>

<Reference URI=**"#Response"**>

<Transforms>

<Transform Algorithm=**"http://www.w3.org/2000/09/xmldsig#enveloped-signature"**/>

<Transform Algorithm=**"http://www.w3.org/2001/10/xml-exc-c14n#"**/>

</Transforms>

<DigestMethod Algorithm=**"http://www.w3.org/2001/04/xmlenc#sha256"**/>

<DigestValue>**WYXOkHAd..........SLOIbwDdHCQk=**</DigestValue>

</Reference>

</SignedInfo>

<SignatureValue>**Bnr1bB8Ic..........bfAjdVUpA==**</SignatureValue>

<KeyInfo>

<X509Data>

<X509Certificate>**MIIFRjCCB..........gpNT2r23YQ==**</X509Certificate>

</X509Data>

</KeyInfo>

</Signature>

</ns2:RegisterInvoiceResponse>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

### DATA EXPORT AND UPLOAD

When the taxpayer is operating in the area where there is no internet connection, he can use the alternative way of invoice registration. Invoice registration requests must be exported into files in a special format. These files can then be uploaded on Central invoice platform which will register invoices. Responses from the registration service will be generated and the taxpayer will download files with responses and bring them back to the ENU for import.

Instead of using the Central invoice platform web application, the taxpayer can bring the files on USB flash drive to the local Tax administration office where the officer will do the same process as in the Central invoice platform.

What taxpayer does in that case is that he uses ENU to creates a file for each of those invoices or cash deposits, and files should be named in <yyyyMMddHHmmSS>\_<TCRCode>\_<IIC>\_request.xml[[1]](#footnote-2) format.

When those .xml files (WS messages) are created, they will be created in the way that only main part of the XML, the one that contains data, will be preserved. It will omit WS header and WS envelope from the message. The file must contain only content of envelope body element. Content of the body element must not be altered to preserve valid signature. To allow time difference between the server and a client, a header part of the message from the body must contain SubseqDelivType attribute.

Then each of those files should be saved in a zipped (archived) folder, named as described in table. Size of the ZIP archive should not exceed 15 MB.

After that the taxpayer transfers that archive to the USB flash drive or other transferable media and takes it to the place where there is an active internet connection. After logging in to Central invoice platform using the certificate, taxpayer imports ZIP archive with invoices represented as XML files. Central invoice platform will extract XML files from the ZIP archive, form a web service message request and send it to fiscalization service. The fiscalization service will make a response, which the Central invoice platform will transform into XML file by extracting the content of SOAP envelope body element.

XML file representing the response will be named <yyyyMMddHHmmSS>\_<IIC>\_response.xml. XML files will be compressed in a ZIP archive with corresponding name <yyyyMMddHHmmSS>\_response.zip where <yyyyMMddHHmmSS> is the same as in the ZIP archive with request XML files. That means that each request in XML file named <yyyyMMddHHmmSS>\_<IIC>\_request.xml or <yyyyMMddHHmmSS>\_deposit\_request.xml will be paired with response in XML file <yyyyMMddHHmmSS>\_<IIC>\_response.xml or <yyyyMMddHHmmSS>\_deposit\_response.xml, and each archive containing XML request named <yyyyMMddHHmmSS>\_request.zip will be paired with archive named <yyyyMMddHHmmSS>\_response.zip containing XML responses.

If there are errors, they will also be stored inside XML responses and the Central Invoice Platform will notify the user about it.

#### XML Example

As mentioned in previous chapter, when there is upload of the invoices, the XML file will be created in a way that only main part of it – the content of the body (without SOAP body element) - the one containing the message – will be preserved, while other parts will be omitted. Header part of the message from the body must contain the SubseqDelivType attribute to allow time difference between the server and a client. Here is the example of a valid request (SOAP envelope elements are removed and, in the example, they are strikethrough):

~~<SOAP-ENV:Envelope xmlns:SOAP-ENV=~~**~~"http://schemas.xmlsoap.org/soap/envelope/"~~**~~>~~

~~<SOAP-ENV:Header/>~~

~~<SOAP-ENV:Body>~~

<RegisterInvoiceRequest xmlns=**"https://efi.tax.gov.me/fs/schema"** xmlns:ns2=**"http://www.w3.org/2000/09/xmldsig#"** Id=**"Request"** Version=**"1"**>

<Header SendDateTime=**"2019-12-05T14:30:13+01:00"** UUID=**"8d216f9a-55bb-445a-be32-30137f11b964"** SubseqDelivType=**"NOINTERNET"**/>

<Invoice BusinUnitCode=**"bb123bb123"** IssueDateTime=**"2019-12-05T14:30:13+01:00"** IIC=**"4AD5A215BEAF85B0416235736A6DACAB"** IICSignaturenvNum=**"1/2019/cc123cc123"** InvOrdNum=**"1"** IsIssuerInVAT=**"true"** IsReverseCharge=**"false"** IsSimplifiedInv=**"false"** OperatorCode=**"oo123oo123"** SoftCode=**"ss123ss123"** TCRCode=**"cc123cc123"** TotPrice=**"20.00"** TotPriceWoVAT=**"16.00"** TotVATAmt=**"4.00"** TypeOfInv=**"CASH"**>

<PayMethods>

<PayMethod Amt=**"20.00"** Type=**"BANKNOTE"**/>

</PayMethods>

<Seller Address=**"Seller address"** Country=**"MNE"** IDNum=**"L91806031N"** IDType=**"JMB"** Name=**"Seller name"** Town=**"Seller town"**/>

<Items>

<I C=**"501234567890"** N=**"Item name"** PA=**"20.00"** PB=**"16.00"** Q=**"1.0"** R=**"0"** RR=**"true"** U=**"piece"** **UPB="16.00"** UPA=**"20.00"** VA=**"4.00"** VR=**"25.00"**/>

</Items>

<SameTaxes>

<SameTax NumOfItems=**"1"** PriceBefVAT=**"16.00"** VATAmt=**"4.00"** VATRate=**"25.00"**/>

</SameTaxes>

</Invoice>

<Signature xmlns=**"http://www.w3.org/2000/09/xmldsig#"**>

<SignedInfo>

<CanonicalizationMethod Algorithm=**"http://www.w3.org/2001/10/xml-exc-c14n#"**/>

<SignatureMethod Algorithm=**"http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"**/>

<Reference URI=**"#Request"**>

<Transforms>

<Transform Algorithm=**"http://www.w3.org/2000/09/xmldsig#enveloped-signature"**/>

<Transform Algorithm=**"http://www.w3.org/2001/10/xml-exc-c14n#"**/>

</Transforms>

<DigestMethod Algorithm=**"http://www.w3.org/2001/04/xmlenc#sha256"**/>

<DigestValue>**ECWby0FKaNQ6..........m8BPjyk6g=**</DigestValue>

</Reference>

</SignedInfo>

<SignatureValue>**LHFUg3xNi..........tjpeYNw==**</SignatureValue>

<KeyInfo>

<X509Data>

<X509Certificate>**MIIFYDCC..........Pifz0UlKJAanmqN3**</X509Certificate>

</X509Data>

</KeyInfo>

</Signature>

</RegisterInvoiceRequest>

~~</SOAP-ENV:Body>~~

~~</SOAP-ENV:Envelope>~~

~~<SOAP-ENV:Envelope xmlns:SOAP-ENV=~~**~~"http://schemas.xmlsoap.org/soap/envelope/"~~**~~>~~

~~<SOAP-ENV:Header/>~~

~~<SOAP-ENV:Body>~~

<RegisterCashDepositRequest xmlns=**"https://efi.tax.gov.me/fs/schema"** xmlns:ns2=**"http://www.w3.org/2000/09/xmldsig#"** Id=**"Request"** Version=**"3"**>

<Header SubseqDelivType=**"NOINTERNET"** SendDateTime=**"2020-07-07T13:57:48+02:00"** UUID=**"f09fd967-ac7e-4f23-8fc4-c72bbca27862"**/>

<CashDeposit CashAmt=**"2000.00"** ChangeDateTime=**"2020-07-07T13:57:48+02:00"** IssuerTIN=**"0123456789"** Operation=**"INITIAL"** TCRCode=**"np830ym389"**/>

<Signature xmlns=**"http://www.w3.org/2000/09/xmldsig#"**>

<SignedInfo>

<CanonicalizationMethod Algorithm=**"http://www.w3.org/2001/10/xml-exc-c14n#"**/>

<SignatureMethod Algorithm=**"http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"**/>

<Reference URI=**"#Request"**>

<Transforms>

<Transform Algorithm=**"http://www.w3.org/2000/09/xmldsig#enveloped-signature"**/>

<Transform Algorithm=**"http://www.w3.org/2001/10/xml-exc-c14n#"**/>

</Transforms>

<DigestMethod Algorithm=**"http://www.w3.org/2001/04/xmlenc#sha256"**/>

<DigestValue>**gYmnHdimKvWgEVySr..........6tZ2pm8BaBbIw4=**</DigestValue>

</Reference>

</SignedInfo>

<SignatureValue>**rx5QLeKSqPDtEpvrld/AvNUhraRBH..........4cNREPde1HY9s8oTDH+ds/dsEEZXQ==**</SignatureValue>

<KeyInfo>

<X509Data>

<X509Certificate>**MIIFEDCCAvigAwIBAgICEBgwDQ..........ZQKFzzNnx+QLBYYOVQ==**</X509Certificate>

</X509Data>

</KeyInfo>

</Signature>

</RegisterCashDepositRequest>

~~</SOAP-ENV:Body>~~

~~</SOAP-ENV:Envelope>~~

Here is the example of a valid response (SOAP envelope elements are removed and, in the example, they are strikethrough):

~~<SOAP-ENV:Envelope xmlns:SOAP-ENV=~~**~~"http://schemas.xmlsoap.org/soap/envelope/"~~**~~>~~

~~<SOAP-ENV:Header/>~~

~~<SOAP-ENV:Body>~~

<ns2:RegisterInvoiceResponse Id=**"Response" Version="3"** xmlns:ns2=**"https://efi.tax.gov.me/fs/schema"** xmlns:ns3=**"http://www.w3.org/2000/09/xmldsig#"**>

<ns2:Header RequestUUID=**"8d216f9a-55bb-445a-be32-30137f11b964"** SendDateTime=**"2019-12-05T14:30:15+01:00"** UUID=**"f8bcb5ae-59fb-41ac-9011-f4db86bbce26"**/>

<ns2:FIC>**a592e7ec-9517-4f02-8d54-ac965f679a8c**</ns2:FIC>

<Signature xmlns=**"http://www.w3.org/2000/09/xmldsig#"**>

<SignedInfo>

<CanonicalizationMethod Algorithm=**"http://www.w3.org/2001/10/xml-exc-c14n#"**/>

<SignatureMethod Algorithm=**"http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"**/>

<Reference URI=**"#Response"**>

<Transforms>

<Transform Algorithm=**"http://www.w3.org/2000/09/xmldsig#enveloped-signature"**/>

<Transform Algorithm=**"http://www.w3.org/2001/10/xml-exc-c14n#"**/>

</Transforms>

<DigestMethod Algorithm=**"http://www.w3.org/2001/04/xmlenc#sha256"**/>

<DigestValue>**WYXOkHAd..........SLOIbwDdHCQk=**</DigestValue>

</Reference>

</SignedInfo>

<SignatureValue>**Bnr1bB8Ic..........bfAjdVUpA==**</SignatureValue>

<KeyInfo>

<X509Data>

<X509Certificate>**MIIFRjCCB..........gpNT2r23YQ==**</X509Certificate>

</X509Data>

</KeyInfo>

</Signature>

</ns2:RegisterInvoiceResponse>

~~</SOAP-ENV:Body>~~

~~</SOAP-ENV:Envelope>~~

~~<SOAP-ENV:Envelope xmlns:SOAP-ENV=~~**~~"http://schemas.xmlsoap.org/soap/envelope/"~~**~~>~~

~~<SOAP-ENV:Header/>~~

~~<SOAP-ENV:Body>~~

<ns2:RegisterCashDepositResponse Id=**"Response" Version="1"** xmlns:ns2=**"https://efi.tax.gov.me/fs/schema"** xmlns:ns3=**"http://www.w3.org/2000/09/xmldsig#"**>

<ns2:Header RequestUUID=**"8d216f9a-55bb-445a-be32-30137f11b964"** SendDateTime=**"2019-12-05T14:30:15+01:00"** UUID=**"f8bcb5ae-59fb-41ac-9011-f4db86bbce26"**/>

<ns2:FCDC>**a592e7ec-9517-4f02-8d54-ac965f679a8c**</ns2:FCDC>

<Signature xmlns=**"http://www.w3.org/2000/09/xmldsig#"**>

<SignedInfo>

<CanonicalizationMethod Algorithm=**"http://www.w3.org/2001/10/xml-exc-c14n#"**/>

<SignatureMethod Algorithm=**"http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"**/>

<Reference URI=**"#Response"**>

<Transforms>

<Transform Algorithm=**"http://www.w3.org/2000/09/xmldsig#enveloped-signature"**/>

<Transform Algorithm=**"http://www.w3.org/2001/10/xml-exc-c14n#"**/>

</Transforms>

<DigestMethod Algorithm=**"http://www.w3.org/2001/04/xmlenc#sha256"**/>

<DigestValue>**WYXOkHAd..........SLOIbwDdHCQk=**</DigestValue>

</Reference>

</SignedInfo>

<SignatureValue>**Bnr1bB8Ic..........bfAjdVUpA==**</SignatureValue>

<KeyInfo>

<X509Data>

<X509Certificate>**MIIFRjCCB..........gpNT2r23YQ==**</X509Certificate>

</X509Data>

</KeyInfo>

</Signature>

</ns2:RegisterCashDepositResponse>

~~</SOAP-ENV:Body>~~

~~</SOAP-ENV:Envelope>~~

Here is an error response example:

<SOAP-ENV:Envelope xmlns:SOAP-ENV=**"http://schemas.xmlsoap.org/soap/envelope/"**>

<SOAP-ENV:Header/>

<SOAP-ENV:Body>

<SOAP-ENV:Fault>

<faultcode>**SOAP-ENV:CLIENT**</faultcode>

<faultstring xml:lang=**"en"**>**Validation failed with digest wrong.**</faultstring>

<detail>

<code>**30**</code>

<requestUUID>**8d216f9a-55bb-445a-be32-30137f11b964**</requestUUID>

<responseUUID>**9416dcca-499a-4724-933d-40d115ea4fc7**</requestUUID>

</detail>

</SOAP-ENV:Fault>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

### QR CODE GENERATION

Each invoice should contain a QR-code which has a URL inside. The URL leads to the web application “Invoice check” and displays information about the invoice if the invoice is successfully registered or instructs user to report an issue if it is not registered within the required timeframe.

QR-code contains a URL with special query string which identifies the invoice. First part of the URL is fixed and contains the protocol, hostname and path followed by the query parameters.

Fixed part of the URL:

* Test environment:

o <https://efitest.tax.gov.me/ic>/#/verify

* Production environment:

o <https://mapr.tax.gov.me/ic/#/verify>

Parameters:

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Description** | **Pattern value** |
| JIKR (*iic*) | Invoice identification code | EA26D5BE7F45827026108F825A8A512B |
| PIB/JMB (*tin*) | Taxpayers tax identification number | 91806031 |
| Created (*crtd*) | Invoice issuing date and time, showin in „yyyy-mm- ddThh:mi:ss+/-zh:zm“ format. | 2019-09-26T13:50:13+01:00 |
| Ord. No. (*ord*) | Invoice ordinal number | 6 |
| PJ (*bu*) | Business unit code | bg517kw842 |
| ENU code (*cr*) | ENU | xb131ap287 |
| Software (*sw*) | ENU software code | gz434bv927 |
| Price (*prc*) | Total invoice price | 199.00 |

Table 140

QR code must be created with at least M error correction level that guaranties that at least 15% of character can be restored.

#### QR code example

Example URL encoded in a QR-code with sample values from the table 143:

<https://efitest.tax.gov.me/ic/#/verify?iic=EA26D5BE7F45827026108F825A8A512B&tin=91806031&crtd=2019-09-26T13:50:13+01:00&ord=6&bu=bg517kw842&cr=xb131ap287&sw=gz434bv927&prc=199.00>



Image 2: QR code example

## ERROR MESSAGE

In case of an error in message request process, error message is sent as an response from the CIS. Messages on error have the same general format based on SOAP message on error, version 1.1, and is expanded with an XML code element that represents the error code number.

### XML FORMAT

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | | | **XML tag** | **Occurrence**  **(min, max)** | **Description** |
| Error | | | Fault | [1, 1] | Root element representing error message. |
|  | Error code | | FaultCode | [1, 1] | XML element representing error class. |
|  | Error explanation | | FaultString | [1, 1] | XML element with the error explanation. |
|  | Details | | Detail | [1, 1] | XML element that carries error messages. |
|  |  | Response UUID | ResponseUUID | [1, 1] | XML element that creates response UUID. |
|  |  | Request UUID | RequestUUID | [0, 1] | XML element that creates request UUID. |
|  |  | Code | Code | [1, 1] | XML element that describes the error with a numeric code. |

Table 141

#### Header

XML element representing the message header.

#### Header: UUID

Element representing the unique universal identifier of the message.

|  |  |
| --- | --- |
| Data type | String |
| Length | 36 chars |
| Pattern | [0-9a-FA-F]{8}–[0-9a-FA-F]{4}–[1-5][0-9a-FA-F]{3}–[89abAB][0-9a-FA-F]{3}–[0-9a-FA-F]{12} |
| Example | 58e0a7d7-eebc-41d8-9669-0800200c9a66  58E0A7D7-EEBC-41D8-9669-0800200C9A66 |

Table 142

#### Error

XML element that shows only if error occurs.

#### Error code5

This is the XML element that represents the error that happened.

|  |  |
| --- | --- |
| Data type | String |
| Value | ENUMERATION |
| Example | CLIENT |

Table 143

Values for error code5 are shown in table below.

|  |  |
| --- | --- |
| **Value** | **Description** |
| CLIENT | Received message is not valid or has incorrect data |
| SERVER | Server problems occurred so message was not processed. |

Table 144

#### Error explanation

Element that contains textual explanation of the error.

|  |  |
| --- | --- |
| Data type | String |
| Length | Undefined |
| Example | Validation failed with digest wrong. |

Table 145

#### Details

XML element that carries numeric error code.

#### Code

Helper element that describes the error code. Numeric error codes are shown in chapter **Pogreška! Izvor reference nije pronađen.**.

|  |  |
| --- | --- |
| Data type | Integer |
| Length | 3 chars |
| Pattern | [1-9][0-9]{0,2} |
| Example | 21 |

Table 146

#### Response UUID

This is the helper element that shows UUID of this error message.

|  |  |
| --- | --- |
| Data type | String |
| Length | 36 chars |
| Pattern | [0-9a-FA-F]{8}–[0-9a-FA-F]{4}–[1-5][0-9a-FA-F]{3}–[89abAB][0-9a-FA-F]{3}–[0-9a-FA-F]{12} |
| Example | 58e0a7d7-eebc-41d8-9669-0800200c9a66  58E0A7D7-EEBC-41D8-9669-0800200C9A66 |

Table 147

#### Request UUID

This is the helper element that shows UUID of this error message.

|  |  |
| --- | --- |
| Data type | String |
| Length | chars |
| Pattern | [0-9a-FA-F]{8}–[0-9a-FA-F]{4}–[1-5][0-9a-FA-F]{3}–[89abAB][0-9a-FA-F]{3}–[0-9a-FA-F]{12} |
| Example | 58e0a7d7-eebc-41d8-9669-0800200c9a66  58E0A7D7-EEBC-41D8-9669-0800200C9A66 |

Table 148

### ERROR CODES

Table below contains the list of error codes that the system can return to the taxpayer. These are not all error codes.

|  |  |  |
| --- | --- | --- |
| **Error code** | **Error origin** | **Error description** |
| 0 | Client | Exception occurred while extracting received XML message during size check. |
| 1 | Client | Received XML message exceed allowed size. |
| 2 | Client | Client time differs from a server’s time by more than allowed time in minutes |
| 10 | Client | Exception occurred while extracting received XML message during XML validation against the XSD. |
| 11 | Client | Received XML message failed XSD validation. |
| 20 | Client | Exception occurred while extracting received XML message during signature check. |
| 21 | Client | Received XML message missing Signature XML element. |
| 22 | Client | Received XML message missing Request XML element. |
| 23 | Client | Exception occurred while extracting Signature XML element during signature check. |
| 24 | Client | Provided more than one Signature XML element. |
| 25 | Client | Signed wrong XML element. |
| 26 | Client | Wrong signature method specified. |
| 27 | Client | Wrong canonicalization method specified. |
| 28 | Client | Wrong digest method specified. |
| 29 | Client | Cryptographic signature wrong. |
| 30 | Client | Digest calculation wrong. |
| 31 | Client | Overall signature wrong. |
| 32 | Client | There are more key info elements than needed. |
| 33 | Client | Certificate provided is not of X509 type of certificate. |
| 34 | Client | Certificate provided is not valid. |
| 35 | Client | Certificate is not issued by registered CA. |
| 36 | Client | Certificate has expired. |
| 37 | Client | Compare the PIB/TIN in XML with the PIB/TIN in the certificate |
| 38 | Client | Certificate status revoked |
| 39 | Client | Certificate status unknown |
| 40 | Client | Invoice amount too large for cash invoice |
| 41 | Client | Business unit code doesn’t reference active business unit (premise) of the taxpayer. |
| 42 | Client | Software code doesn’t references active software. |
| 43 | Client | Maintainer code doesn’t references active maintainer. |
| 44 | Client | Issuer VAT status doesn’t correspond to the IsIssuerInVAT attribute. |
| 45 | Client | Valid from cannot be in the past. |
| 46 | Client | Valid to cannot be in the past. |
| 47 | Client | Valid to cannot be before Valid from. |
| 48 | Client | Active ENU cannot be updated. |
| 49 | Client | Change date and time differences from CISs time more than allowed time in minutes. |
| 50 | Client | Cash amount cannot be negative for INITIAL operation. |
| 51 | Client | Cash amount cannot be zero for WITHDRAW operation. |
| 52 | Client | Taxpayer doesn’t exist in the Registry of taxpayers. |
| 53 | Client | ENU code doesn’t reference registered or active ENU or the ENU doesn’t belongs to the referenced issuer. |
| 54 | Client | Issuer ID type must be PIB/TIN. |
| 55 | Client | Taxpayer is not active in the Registry of taxpayers. |
| 56 | Client | Cash deposit with operation INITIAL already registered for a current day. |
| 57 | Client | Deactivated ENU cannot be modified. |
| 9xx | Server | Exceptions for internal server. |

Table 149

### XML EXAMPLE

<SOAP-ENV:Envelope xmlns:SOAP-ENV=**"http://schemas.xmlsoap.org/soap/envelope/"**>

<SOAP-ENV:Header/>

<SOAP-ENV:Body>

<SOAP-ENV:Fault>

<faultcode>**SOAP-ENV:CLIENT**</faultcode>

<faultstring xml:lang=**"en"**>**OPIS GREŠKE.**</faultstring>

<detail>

<code>**KOD GREŠKE**</code>

<requestUUID>**78dde160-2b33-40e4-98fa-f6a2c34475a3**</requestUUID>

<responseUUID>**9416dcca-499a-4724-933d-40d115ea4fc7**</requestUUID>

</detail>

</SOAP-ENV:Fault>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

# SECURITY

Following chapter describes security principles used by fiscalization service and parties that communicate with it.

## DIGITAL CERTIFICATES

For the purpose of the securing fiscalization service and identifying clients that consumes the service, three types of certificates shall be issued:

* Certificate for transport security with common name issued to the service.

Used to secure transport between the service and an issuer.

Issued by public CA different from registered CA.

* Certificate for message security issued to the service.

Used to digitally sign response data messages.

Issued registered CA.

* Certificate for message security issued to an issuer.

Used to digitally sign request data messages and IKOF data elements.

Issued by registered CA.

Digital confirmation for fiscalization needs is provided by registered CA. Details for getting a certificate is described and defined by registered CA themselves.

## TRANSPORT SECURITY

To ensure data security and integrity of the communication between the issuer and the service, service is using One-way TLS, specifically protocol version TLS V1.2. Service presents a certificate to client issued by the registered CA.

## MESSAGE SECURITY

To ensure unambiguous identification of the taxpayer and to provide unchanged content, each request data message and IIC data element is digitally signed with a private key that is unique pair with the valid taxpayer certificate. Response data messages form the CIS are digitally signed with a private key that is unique pair with the valid CIS certificate.

In most cases, the private key used to digitally sign request data message and IKOF data is the same. An exception under this rule is possible if the certificate used at the time of initial creation of the request is no longer valid at the time of resending the request. In that case, a private key from valid corresponding certificate must be used to digitally sign request data message, but not the IIC data element, which remains the same.

Request and response data messages are digitally signed according to the XML Signature Syntax and Processing standard edition 1.1 available at <https://www.w3.org/TR/xmldsig-core/>. Additional description is available in the chapter **Pogreška! Izvor reference nije pronađen.**.

IIC data element is created and digitally signed according to the custom cryptographic algorithm described in the chapter 4.3.2.

### ELECTRONIC REQUEST AND RESPONS MESSAGE SIGNING

Every request and response data message described in the chapter **Pogreška! Izvor reference nije pronađen.**, must contain signature XML element. That element is generated according to XML Signature Syntax and Processing standard edition 1.1 available at <https://www.w3.org/TR/xmldsig-core/>.

Element to signed is a first and only element inside soap envelope body XML element, with Id equals to Request or Response, depending on the message direction.

XML digital signature element is created with following options:

* Signature type: Enveloped, <http://www.w3.org/2000/09/xmldsig#enveloped-signature>
* Canonicalization method: C14 Exclusive, [http://www.w3.org/2001/10/xml-exc-c14n#](http://www.w3.org/2001/10/xml-exc-c14n)
* Digest method: SHA256, <http://www.w3.org/2001/04/xmlenc#sha256>
* Signing method: RSA SHA256, <http://www.w3.org/2001/04/xmldsig-more#rsa-sha256>

### IKOF DATA ELEMENT

IKOF, the identification code is the alphanumeric security code that is generated by the invoice issuer, and which uniquely matches the issued invoice. It is generated by concatenating the specific invoice elements and it is signed by the issuer private key.

IKOF has two purposes:

1. To protect the issuer from malicious third party, because the issuer is the only one who can restore it by inserting the same parameters and using the same key
2. To check if the issued invoice is registered in the CIS.

On Tax administration request, the taxpayer, based on in parameters, must generate the IKOF as same as the invoice value.

IKOF is generated by these algorithm steps:

1. Concatenating parameters
2. Digital signature calculation with SHA256, RSA and RSASSA-PKCS-v1\_5 addition
3. Summary calculation

#### Parameters concatenation

IKOF is generated by concatenating these parameters:

* JMB/PIB of the issuer (Chapter 3.7.1.53)
* Issuing date and time (Chapter 3.7.1.9)
* Invoice ordinal number (Chapter 3.7.1.11)
* Business unit code (Chapter 3.7.1.21)
* ENU code (Chapter 3.7.1.12)
* Software code (Chapter 3.7.1.22)
* Total price (Chapter 3.7.1.19)

Before concatenation, all parameters must be converted into UTF-8 encoding. Parameters are concatenated with pipe character UTF-8 with decimal code 124.

For example, for parameters:

* JMB/PIB of the issuer: 12345678
* Issuing date and time: 2019-06-12T17:05:43+02:00
* Invoice ordinal number: 9952
* Business unit code: bb123bb123
* ENU code: cc123cc123
* Software code: ss123ss123
* Total price: 99.01

Resulting value is:

12345678|2019-06-12T17:05:43+02:00|9952|bb123bb1231|cc123cc1231|ss123ss123|99.01

#### Digital signature calculation

After calculating, resulting value is hashed with the SHA256 algorithm, and then signed by RSA algorithm and issuer’s private key.

For example, for values:

* Total value:

12345678|2019-06-12T17:05:43+02:00|9952|bb123bb123|cc123cc123|ss123ss123|99.01

* Private key coded by PEM:

|  |
| --- |
| -----BEGIN RSA PRIVATE KEY-----  MIIEpAIBAAKCAQEA6zOR5ItNYHJNVMxljZtd/KQUyGIozbnIJ8IWqcEesktRV5FF  HviQZsx2DpyeVQTu/Kel9Xh+Z6OZ6t5sADzfYnkwCrsb0FhT+01m2PIHaIUZhVtc  ppn0gxNWfgzW4sTvTyrYk6O1Kxymsx/rck/WRQB1mp68au8mgGMzGukHfL7Wk4jO  U5VD3HlStBx1MjVW+soN5GUL/rWGaYun6Zsn9aYYEujbOhKvKDy8nOtNIS69dqdd  piZAkvdh9sYdF1ElgXZhdmZsGURMm6OcePUPZO/HFKq7RlK6vIxXVI6l9O6tWt+G  uhul8e0x2VTwbTdpwG4FpdfUTqUDK6cswHOhTQIDAQABAoIBAQCqBWJuUqDBmn76  ULMMlYZwjfAUFpkmdikRTIVzew4EltubMIFF7Sr9lMm2sFLoZKOZ8lrOwqalpqcq  GFT8KwTUO4SWDUIC7wbuf7pcE0F1tdmIBE5KhLozUnRQtFlWHkRb9z4OI+Zf3ttG  W0mpHbtnr/hTqHHN30j2wD7+MfvemPbcAvu9JLCYUzUZ06qxUwAjyFgsW7YyLa0a  qFB0QOYc6RsLvoSFXW0M5ghdtgoZvl+ayt4fgz1L3FjAMuXoLEX/778VA92/NZ0Q  mzQdKTT6B4Pm5s8XrY9OhLlsYqKuyR/aoSHC/anSLw0yJ/5Gis2gmCwo3a7+PEYy  LUN7C0yFAoGBAPhgYufTkdod5PqG/SCEE2i6pjk0ZnuIUu9f2cmhxnvyChlig2wk  oDWUSGuXwItNF+X7j3XoZz8FNJcriK7KP2UPDOWP0ZvxZgZEcmwut27x1vVjzjCG  sl0w5fFO363hhtX35Jq2lVZGbN1LpIoEZgCeS/nBs+9DcRjDoXliKWfHAoGBAPJr  qSWLVO3gIG1wikXBWCYZUTSzsO6NWfxcWPHKTnKVrOifBTK23zuZ6ggluNqLz/Ae  64ZwssMoIVIyXE01XMPP8io4QidyVEd2n70pjrVcUVYyr9IwKmchmNBfKFMof05f  NV29P1Am1Jqv2EQi5jE/BbBu9kLifs2YyGBAn/ZLAoGAVsLsqciZAVVCAFWZJHue  gA37NK5eQja7qcyUuj9dozxIVNe5ytP8dtrmdVccNkzm1TqLwYc+UaBS35+gblZN  0NJyEdqsQMoRdo0AX1PuVb369ds4UnEq6yzClgmUTxwhyqp+W6D+B5YwPxlGT8P7  kam6JnOIlEK9xgXIaStmBU8CgYB6RwXVszcOmYuhyC9mygSNix2j6LNpUJFAMtCG  fZYeRBMobvWvRADLznH21Bgu3HDxXJdOg9AXkklkbZSTOURmXKB43VG5Ffke5t3i  C3E5V6yLPxvieHsa9B5hlG4BrB6yyGFhvBCQfFWnBOWgUL4tvu0+tmmvCRIO4G7J  5i8JiwKBgQCQHTfRrGaEsq1BG7zPOQSqo9q5cxL8WzYd0sTs3FDcwCtHqxBEQ3rr  O/l+HvRa+y6ZEH6q4pREewTIymfv9tmGxVe3f8zrKGR5litvN6OnZuWJdq57Y1lN  J1sdpMxTtxQQmexsADif+QByCvdeFKE5C3veMLdgS5I6HTMN9k5laA==  -----END RSA PRIVATE KEY----- |

Signature value result is:



#### Calculated summary

After signing, the resulting value is signed with hash using MD5 algorithm.

For signature value of:



MD5 summary value is: E4033D471FEEA47A3C664B15C669C709

# ANEX 1 – CODE EXAMPLES

This chapter contains code examples for specific actions.

## IKOF GENERATION CODE

### JAVA EXAMPLE

This is IKOF generation example in Java language. Values are hardcoded as this is for example purposes only.

**import** java**.**io**.**FileInputStream**;**

**import** java**.**security**.\*;**

**import** javax**.**xml**.**bind**.**DatatypeConverter**;**

public class SampleGenerateIIC **{**

private static final String KEYSTORE\_LOCATION **=** "\*\*\*.p12"**;**

private static final String KEYSTORE\_TYPE **=** "PKCS12"**;**

private static final String KEYSTORE\_PASS **=** "\*\*\*"**;**

private static final String KEYSTORE\_KEY\_ALIAS **=** "\*\*\*"**;**

public static void main**(**String**[]** args**)** **{**

String iicInput **=** ""**;**

// issuerTIN

iicInput **+=** "12345678"**;**

// dateTimeCreated

iicInput **+=** "|2019-06-12T17:05:43+02:00"**;**

// invoiceNumber

iicInput **+=** "|9952"**;**

// busiUnitCode

iicInput **+=** "|bb123bb123"**;**

// tcrCode

iicInput **+=** "|cc123cc123"**;**

// softCode

iicInput **+=** "|ss123ss123"**;**

// totalPrice

iicInput **+=** "|99.01"**;**

**try** **(**FileInputStream fileInputStream **=** **new** FileInputStream**(**KEYSTORE\_LOCATION**))** **{**

// Load a private from a key store

KeyStore keyStore **=** KeyStore**.**getInstance**(**KEYSTORE\_TYPE**);**

keyStore**.**load**(**fileInputStream**,** KEYSTORE\_PASS**.**toCharArray**());**

Key privateKey **=** keyStore**.**getKey**(**KEYSTORE\_KEY\_ALIAS**,** KEYSTORE\_PASS**.**toCharArray**());**

// Create IIC signature according to RSASSA-PKCS-v1\_5

Signature signature **=** Signature**.**getInstance**(**"SHA256withRSA"**);**

signature**.**initSign**((**PrivateKey**)**privateKey**);**

signature**.**update**(**iicInput**.**getBytes**());**

byte**[]** iicSignature **=** signature**.**sign**();**

String iicSignatureString **=** DatatypeConverter**.**printHexBinary**(**iicSignature**).**toUpperCase**();**

System**.**out**.**println **(**"The IIC signature is: " **+** iicSignatureString**);**

// Hash IIC signature with MD5 to create IIC

MessageDigest md **=** MessageDigest**.**getInstance**(**"MD5"**);**

byte**[]** iic **=** md**.**digest**(**iicSignature**);**

String iicString **=** DatatypeConverter**.**printHexBinary**(**iic**).**toUpperCase**();**

System**.**out**.**println **(**"The IIC is: " **+** iicString**);**

**}** **catch(**Exception e**)** **{**

e**.**printStackTrace**();**

**}**

**}**

**}**

### C# EXAMPLE

This is IKOF generation example in C## language. Values are hardcoded as this is for example purposes only.

**using** System**;**

**using** System**.**Security**.**Cryptography**;**

**using** System**.**Security**.**Cryptography**.**X509Certificates**;**

**using** System**.**Text**;**

**namespace** FiscalizationSigningUtilityDotNet

**{**

class SampleGenerateIIC

**{**

**private** const String KEYSTORE\_LOCATION **=** "\*\*\*.p12"**;**

**private** const String KEYSTORE\_PASS **=** "\*\*\*"**;**

**public** static void Main**(**string**[]** args**)**

**{**

String iicInput **=** ""**;**

// issuerTIN

iicInput **+=** "12345678"**;**

// dateTimeCreated

iicInput **+=** "|2019-06-12T17:05:43+02:00"**;**

// invoiceNumber

iicInput **+=** "|9952"**;**

// busiUnitCode

iicInput **+=** "|bb123bb123"**;**

// tcrCode

iicInput **+=** "|cc123cc123"**;**

// softCode

iicInput **+=** "|ss123ss123"**;**

// totalPrice

iicInput **+=** "|99.01"**;**

**using** **(**X509Certificate2 keyStore **=** **new** X509Certificate2**(**KEYSTORE\_LOCATION**,** KEYSTORE\_PASS**))**

**{**

**try**

**{**

// Load a private from a key store

RSA privateKey **=** keyStore**.**GetRSAPrivateKey**();**

// Create IIC signature according to RSASSA-PKCS-v1\_5

byte**[]** iicSignature **=** privateKey**.**SignData**(**Encoding**.**ASCII**.**GetBytes**(**iicInput**),** HashAlgorithmName**.**SHA256**,** RSASignaturePadding**.**Pkcs1**);**

string iicSignatureString **=** BitConverter**.**ToString**(**iicSignature**).**Replace**(**"-"**,** string**.**Empty**);**

Console**.**WriteLine**(**"The IIC signature is: " **+** iicSignatureString**);**

// Hash IIC signature with MD5 to create IIC

byte**[]** iic **=** **((**HashAlgorithm**)**CryptoConfig**.**CreateFromName**(**"MD5"**)).**ComputeHash**(**iicSignature**);**

string iicString **=** BitConverter**.**ToString**(**iic**).**Replace**(**"-"**,** string**.**Empty**);**

Console**.**WriteLine**(**"The IIC is: " **+** iicString**);**

**}**

**catch** **(**Exception ex**)**

**{**

Console**.**WriteLine**(**ex**.**Message**);**

**}**

**}**

**}**

**}**

**}**

## SIGNATURE GENERATION CODE

### JAVA EXAMPLE

This is signature generation example in Java language. Values are hardcoded as this is for example purposes only.

**import** java**.**io**.\*;**

**import** java**.**security**.\*;**

**import** java**.**security**.**cert**.**X509Certificate**;**

**import** java**.**util**.\*;**

**import** javax**.**xml**.**crypto**.**dsig**.\*;**

**import** javax**.**xml**.**crypto**.**dsig**.**keyinfo**.\*;**

**import** javax**.**xml**.**crypto**.**dsig**.**spec**.\*;**

**import** javax**.**xml**.**crypto**.**dsig**.**dom**.**DOMSignContext**;**

**import** javax**.**xml**.**parsers**.**DocumentBuilder**;**

**import** javax**.**xml**.**parsers**.**DocumentBuilderFactory**;**

**import** javax**.**xml**.**transform**.\*;**

**import** javax**.**xml**.**transform**.**dom**.**DOMSource**;**

**import** javax**.**xml**.**transform**.**stream**.**StreamResult**;**

**import** org**.**w3c**.**dom**.\*;**

**import** org**.**xml**.**sax**.**InputSource**;**

public class SampleGenerateSignature **{**

private static final XMLSignatureFactory xmlSigFactory **=** XMLSignatureFactory**.**getInstance**(**"DOM"**);**

public static final String XML\_SCHEMA\_NS **=** "https://efi.tax.gov.me/fs/schema"**;**

public static final String XML\_REQUEST\_ELEMENT **=** "RegisterInvoiceRequest"**;**

public static final String XML\_REQUEST\_ID **=** "Request"**;**

public static final String XML\_SIG\_METHOD **=** "http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"**;**

private static final String REQUEST\_TO\_SIGN **=**

"<RegisterInvoiceRequest " **+**

" xmlns=\"https://efi.tax.gov.me/fs/schema\" " **+**

" xmlns:ns2=\"http://www.w3.org/2000/09/xmldsig#\" " **+**

" Id=\"Request\" " **+**

" Version=\"3\">\r\n" **+**

" <Header>...</Header>\r\n" **+**

" <Invoice>...</Invoice>\r\n" **+**

"</RegisterInvoiceRequest>"**;**

private static final String KEYSTORE\_LOCATION **=** "\*\*\*.p12"**;**

private static final String KEYSTORE\_TYPE **=** "PKCS12"**;**

private static final String KEYSTORE\_PASS **=** "\*\*\*"**;**

private static final String KEYSTORE\_KEY\_ALIAS **=** "\*\*\*"**;**

public static void main**(**String**[]** args**)** **{**

**try** **(**FileInputStream fileInputStream **=** **new** FileInputStream**(**KEYSTORE\_LOCATION**))** **{**

// Load a private from a key store

KeyStore keyStore **=** KeyStore**.**getInstance**(**KEYSTORE\_TYPE**);**

keyStore**.**load**(**fileInputStream**,** KEYSTORE\_PASS**.**toCharArray**());**

Key privateKey **=** keyStore**.**getKey**(**KEYSTORE\_KEY\_ALIAS**,** KEYSTORE\_PASS**.**toCharArray**());**

X509Certificate certificate **=** **(**X509Certificate**)**keyStore**.**getCertificate**(**KEYSTORE\_KEY\_ALIAS**);**

// Load XML to DOC

DocumentBuilderFactory docFactory **=** DocumentBuilderFactory**.**newInstance**();**

docFactory**.**setNamespaceAware**(true);**

DocumentBuilder docBuilder **=** docFactory**.**newDocumentBuilder**();**

Document doc **=** docBuilder**.**parse**(new** InputSource**(new** StringReader**(**REQUEST\_TO\_SIGN**)));**

// Find root request element

NodeList nodeToSignList **=** doc**.**getElementsByTagNameNS**(**XML\_SCHEMA\_NS**,** XML\_REQUEST\_ELEMENT**);**

**if** **(**nodeToSignList**.**getLength**()** **==** 0**)** **{**

**throw** **new** Exception**(**String**.**format**(**"XML element %s not found"**,** XML\_REQUEST\_ELEMENT**));**

**}**

Node nodeToSign **=** nodeToSignList**.**item**(**0**);**

// Create transform list

List**<**Transform**>** transformList **=** **new** ArrayList**<>();**

transformList**.**add**(**xmlSigFactory**.**newTransform**(**Transform**.**ENVELOPED**,** **(**TransformParameterSpec**)** **null));**

transformList**.**add**(**xmlSigFactory**.**newTransform**(**CanonicalizationMethod**.**EXCLUSIVE**,** **(**C14NMethodParameterSpec**)** **null));**

// Create digest reference element

Reference ref **=** xmlSigFactory**.**newReference**(**

"#" **+** XML\_REQUEST\_ID**,**

xmlSigFactory**.**newDigestMethod**(**DigestMethod**.**SHA256**,** **null),**

transformList**,**

**null,**

**null);**

// Create signature method

SignatureMethod signatureMethod **=** xmlSigFactory**.**newSignatureMethod**(**XML\_SIG\_METHOD**,** **(**SignatureMethodParameterSpec**)** **null);**

// Create signed info element

SignedInfo signedInfo **=** xmlSigFactory**.**newSignedInfo**(**

xmlSigFactory**.**newCanonicalizationMethod**(**CanonicalizationMethod**.**EXCLUSIVE**,** **(**C14NMethodParameterSpec**)** **null),**

signatureMethod**,**

Collections**.**singletonList**(**ref**));**

// Add certificate

List**<**X509Certificate**>** certificateList **=** **new** ArrayList**<>();**

certificateList**.**add**(**certificate**);**

// Create key info element

KeyInfoFactory keyInfoFactory **=** xmlSigFactory**.**getKeyInfoFactory**();**

X509Data x509Data **=** keyInfoFactory**.**newX509Data**(**certificateList**);**

KeyInfo keyInfo **=** keyInfoFactory**.**newKeyInfo**(**Collections**.**singletonList**(**x509Data**));**

// Create context for signing

DOMSignContext dsc **=** **new** DOMSignContext**(**privateKey**,** nodeToSign**);**

dsc**.**setIdAttributeNS**((**Element**)**nodeToSign**,** **null,** "Id"**);**

// Sign document

XMLSignature signature **=** xmlSigFactory**.**newXMLSignature**(**signedInfo**,** keyInfo**);**

signature**.**sign**(**dsc**);**

// Output to string

TransformerFactory transformFactory **=** TransformerFactory**.**newInstance**();**

Transformer transformer **=** transformFactory**.**newTransformer**();**

transformer**.**setOutputProperty**(**OutputKeys**.**OMIT\_XML\_DECLARATION**,** "yes"**);**

StringWriter sw **=** **new** StringWriter**();**

StreamResult streamRes **=** **new** StreamResult**(**sw**);**

transformer**.**transform**(new** DOMSource**(**doc**),** streamRes**);**

System**.**out**.**println **(**"Signed document is: " **+** sw**.**toString**());**

**}** **catch** **(**Exception e**)** **{**

e**.**printStackTrace**();**

**}**

**}**

**}**

### C# EXAMPLE

This is signature generation example in C# language. Values are hardcoded as this is for example purposes only.

**using** System**;**

**using** System**.**IO**;**

**using** System**.**Security**.**Cryptography**;**

**using** System**.**Security**.**Cryptography**.**X509Certificates**;**

**using** System**.**Security**.**Cryptography**.**Xml**;**

**using** System**.**Xml**;**

**namespace** FiscalizationSigningUtilityDotNet

**{**

class SampleGenerateSignature

**{**

**public** const String XML\_SCHEMA\_NS **=** "https://efi.tax.gov.me/fs/schema"**;**

**public** const String XML\_REQUEST\_ID **=** "Request"**;**

**public** const String XML\_SIG\_METHOD **=** "http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"**;**

**public** const String XML\_DIG\_METHOD **=** "http://www.w3.org/2001/04/xmlenc#sha256"**;**

**private** const String REQUEST\_TO\_SIGN **=**

"<RegisterInvoiceRequest " **+**

" xmlns=\"https://efi.tax.gov.me/fs/schema\" " **+**

" xmlns:ns2=\"http://www.w3.org/2000/09/xmldsig#\" " **+**

" Id=\"Request\" " **+**

" Version=\"3\">\r\n" **+**

" <Header>...</Header>\r\n" **+**

" <Invoice>...</Invoice>\r\n" **+**

"</RegisterInvoiceRequest>"**;**

**private** const String KEYSTORE\_LOCATION **=** "\*\*\*.p12"**;**

**private** const String KEYSTORE\_PASS **=** "\*\*\*"**;**

**public** static void Main**(**string**[]** args**)**

**{**

**using** **(**X509Certificate2 keyStore **=** **new** X509Certificate2**(**KEYSTORE\_LOCATION**,** KEYSTORE\_PASS**))**

**{**

**try**

**{**

// Load a private from a key store

RSA privateKey **=** keyStore**.**GetRSAPrivateKey**();**

// Convert string XML to object

XmlDocument request **=** **new** XmlDocument**();**

request**.**LoadXml**(**REQUEST\_TO\_SIGN**);**

// Create key info element

KeyInfo keyInfo **=** **new** KeyInfo**();**

KeyInfoX509Data keyInfoData **=** **new** KeyInfoX509Data**();**

keyInfoData**.**AddCertificate**(**keyStore**);**

keyInfo**.**AddClause**(**keyInfoData**);**

// Create signature reference

Reference reference **=** **new** Reference**(**""**);**

reference**.**AddTransform**(new** XmlDsigEnvelopedSignatureTransform**(false));**

reference**.**AddTransform**(new** XmlDsigExcC14NTransform**(false));**

reference**.**DigestMethod **=** XML\_DIG\_METHOD**;**

reference**.**Uri **=** "#" **+** XML\_REQUEST\_ID**;**

// Create signature

SignedXml xml **=** **new** SignedXml**(**request**);**

xml**.**SigningKey **=** privateKey**;**

xml**.**SignedInfo**.**CanonicalizationMethod **=** SignedXml**.**XmlDsigExcC14NTransformUrl**;**

xml**.**SignedInfo**.**SignatureMethod **=** XML\_SIG\_METHOD**;**

xml**.**KeyInfo **=** keyInfo**;**

xml**.**AddReference**(**reference**);**

xml**.**ComputeSignature**();**

// Add signature element to the request

XmlElement signature **=** xml**.**GetXml**();**

request**.**DocumentElement**.**AppendChild**(**signature**);**

// Convert signed request to string and print

StringWriter sw **=** **new** StringWriter**();**

XmlTextWriter xw **=** **new** XmlTextWriter**(**sw**);**

request**.**WriteTo**(**xw**);**

Console**.**WriteLine**(**"Signed document is: " **+** sw**.**ToString**());**

**}**

**catch** **(**Exception ex**)**

**{**

Console**.**WriteLine**(**ex**.**Message**);**

**}**

**}**

**}**

**}**

**}**

# ANEX 2 – WSDL version 1

|  |
| --- |
| <?xml version=**"1.0"** encoding=**"UTF-8"**?>  <wsdl:definitions  name=**"FiscalizationService"**  targetNamespace=**"https://efi.tax.gov.me/fs"**  xmlns:me=**"https://efi.tax.gov.me/fs"**  xmlns:wsdl=**"http://schemas.xmlsoap.org/wsdl/"**  xmlns:soap=**"http://schemas.xmlsoap.org/wsdl/soap/"**  xmlns:xsd=**"http://www.w3.org/2001/XMLSchema"**  xmlns:mes=**"https://efi.tax.gov.me/fs/schema"**  xmlns:xsi=**"http://www.w3.org/2001/XMLSchema-instance"**>  <wsdl:types>  <xsd:schema>  <xsd:import namespace=**"https://efi.tax.gov.me/fs/schema"** schemaLocation=**"eficg-fiscalization-service.xsd"**/>  </xsd:schema>  </wsdl:types>  <wsdl:message name=**"RegisterInvoiceRequest"**>  <wsdl:documentation>**Element representing register sale message.**</wsdl:documentation>  <wsdl:part element=**"mes:RegisterInvoiceRequest"** name=**"request"** />  </wsdl:message>  <wsdl:message name=**"RegisterInvoiceResponse"**>  <wsdl:documentation>**Element representing register invoice response message.**</wsdl:documentation>  <wsdl:part element=**"mes:RegisterInvoiceResponse"** name=**"response"** />  </wsdl:message>    <wsdl:message name=**"RegisterTCRRequest"**>  <wsdl:documentation>**Element representing request of the TCR registration message.**</wsdl:documentation>  <wsdl:part element=**"mes:RegisterTCRRequest"** name=**"request"** />  </wsdl:message>    <wsdl:message name=**"RegisterTCRResponse"**>  <wsdl:documentation>**Element representing response of the TCR registration message.**</wsdl:documentation>  <wsdl:part element=**"mes:RegisterTCRResponse"** name=**"response"** />  </wsdl:message>    <wsdl:message name=**"RegisterCashDepositRequest"**>  <wsdl:documentation>**Element representing request of the cash deposit request message.**</wsdl:documentation>  <wsdl:part element=**"mes:RegisterCashDepositRequest"** name=**"request"** />  </wsdl:message>    <wsdl:message name=**"RegisterCashDepositResponse"**>  <wsdl:documentation>**Element representing response of the cash deposit response message.**</wsdl:documentation>  <wsdl:part element=**"mes:RegisterCashDepositResponse"** name=**"response"** />  </wsdl:message>    <wsdl:portType name=**"FiscalizationServicePortType"**>  <wsdl:operation name=**"registerInvoice"**>  <wsdl:input message=**"me:RegisterInvoiceRequest"**/>  <wsdl:output message=**"me:RegisterInvoiceResponse"**/>  </wsdl:operation>  <wsdl:operation name=**"registerTCR"**>  <wsdl:input message=**"me:RegisterTCRRequest"**/>  <wsdl:output message=**"me:RegisterTCRResponse"**/>  </wsdl:operation>  <wsdl:operation name=**"registerCashDeposit"**>  <wsdl:input message=**"me:RegisterCashDepositRequest"**/>  <wsdl:output message=**"me:RegisterCashDepositResponse"**/>  </wsdl:operation>  </wsdl:portType>  <wsdl:binding name=**"FiscalizationServiceSoap"** type=**"me:FiscalizationServicePortType"**>  <soap:binding style=**"document"** transport=**"http://schemas.xmlsoap.org/soap/http"**/>  <wsdl:operation name=**"registerInvoice"**>  <soap:operation soapAction=**"https://efi.tax.gov.me/fs/RegisterInvoice"**/>  <wsdl:input>  <soap:body use=**"literal"**/>  </wsdl:input>  <wsdl:output>  <soap:body use=**"literal"**/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name=**"registerTCR"**>  <soap:operation soapAction=**"https://efi.tax.gov.me/fs/RegisterTCR"**/>  <wsdl:input>  <soap:body use=**"literal"**/>  </wsdl:input>  <wsdl:output>  <soap:body use=**"literal"**/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name=**"registerCashDeposit"**>  <soap:operation soapAction=**"https://efi.tax.gov.me/fs/RegisterCashDeposit"**/>  <wsdl:input>  <soap:body use=**"literal"**/>  </wsdl:input>  <wsdl:output>  <soap:body use=**"literal"**/>  </wsdl:output>  </wsdl:operation>  </wsdl:binding>  <wsdl:service name=**"FiscalizationService"**>  <wsdl:port name=**"FiscalizationServicePort"** binding=**"me:FiscalizationServiceSoap"**>  <soap:address location=**"https://efi.tax.gov.me/fs-v1"**/>  </wsdl:port>  </wsdl:service>  </wsdl:definitions> |

# ANEX 3 – XSD version 1

|  |
| --- |
| <?xml version=**"1.0"** encoding=**"UTF-8"**?>  <schema  targetNamespace=**"https://efi.tax.gov.me/fs/schema"**  xmlns:me=**"https://efi.tax.gov.me/fs/schema"**  xmlns=**"http://www.w3.org/2001/XMLSchema"**  xmlns:ds=**"http://www.w3.org/2000/09/xmldsig#"**  xmlns:vc=**"http://www.w3.org/2007/XMLSchema-versioning"**  elementFormDefault=**"qualified"**  vc:minVersion=**"1.1"**>  <import namespace=**"http://www.w3.org/2000/09/xmldsig#"** schemaLocation=**"xmldsig-core-schema.xsd"**/>  <element name=**"RegisterInvoiceRequest"**>  <annotation>  <documentation>**Root XML element representing register invoice message.**</documentation>  </annotation>  <complexType>  <all minOccurs=**"1"** maxOccurs=**"1"**>  <element name=**"Header"** type=**"me:RegisterInvoiceRequestHeaderType"** minOccurs=**"1"** maxOccurs=**"1"**>  <annotation>  <documentation>**XML element representing header of the invoice containing data about the message (request) sent.**</documentation>  </annotation>  </element>  <element name=**"Invoice"** type=**"me:InvoiceType"** minOccurs=**"1"** maxOccurs=**"1"**>  <annotation>  <documentation>**XML element representing a single invoice.**</documentation>  </annotation>  </element>  <element ref=**"ds:Signature"** minOccurs=**"1"** maxOccurs=**"1"**>  <annotation>  <documentation>**XML element representing signature for the invoice.**</documentation>  </annotation>  </element>  </all>  <attribute name=**"Id"** type=**"string"** use=**"required"** fixed=**"Request"**>  <annotation>  <documentation>**Attribute used for signature creation and verification.**</documentation>  </annotation>  </attribute>  <attribute name=**"Version"** type=**"me:IntSType"** use=**"required"** fixed=**"1"**>  <annotation>  <documentation>**Attribute used to specify compliance with XSD schema.**</documentation>  </annotation>  </attribute>  </complexType>  </element>  <element name=**"RegisterInvoiceResponse"**>  <annotation>  <documentation>**Root XML element representing register invoice response message.**</documentation>  </annotation>  <complexType>  <all>  <element name=**"Header"** type=**"me:RegisterInvoiceResponseHeaderType"** minOccurs=**"1"** maxOccurs=**"1"**>  <annotation>  <documentation>**Element representing the header of the message.**</documentation>  </annotation>  </element>  <element name=**"FIC"** type=**"me:UUIDSType"** minOccurs=**"1"** maxOccurs=**"1"**>  <annotation>  <documentation>**CIS generated verification code that can be used to uniquely identify registered invoice.**</documentation>  </annotation>  </element>  <element ref=**"ds:Signature"** minOccurs=**"1"** maxOccurs=**"1"**/>  </all>  <attribute name=**"Id"** type=**"string"** use=**"required"** fixed=**"Response"**>  <annotation>  <documentation>**Identification of the response, used to reference a signature.**</documentation>  </annotation>  </attribute>  <attribute name=**"Version"** type=**"me:IntSType"** use=**"required"** fixed=**"1"**>  <annotation>  <documentation>**Identification of the schema version.**</documentation>  </annotation>  </attribute>  </complexType>  </element>  <element name=**"RegisterCashDepositRequest"**>  <annotation>  <documentation>**Root XML element representing cash deposit request message.**</documentation>  </annotation>  <complexType>  <all minOccurs=**"1"** maxOccurs=**"1"**>  <element name=**"Header"** type=**"me:RegisterCashDepositRequestHeaderType"** minOccurs=**"1"** maxOccurs=**"1"**>  <annotation>  <documentation>**Element representing the header of the message.**</documentation>  </annotation>  </element>  <element name=**"CashDeposit"** type=**"me:CashDepositType"** minOccurs=**"1"** maxOccurs=**"1"**>  <annotation>  <documentation>**Element representing a single cash deposit request.**</documentation>  </annotation>  </element>  <element ref=**"ds:Signature"** minOccurs=**"1"** maxOccurs=**"1"**/>  </all>  <attribute name=**"Id"** type=**"string"** use=**"required"** fixed=**"Request"**>  <annotation>  <documentation>**Identification of the request, used to reference a signature.**</documentation>  </annotation>  </attribute>  <attribute name=**"Version"** type=**"me:IntSType"** use=**"required"** fixed=**"1"**>  <annotation>  <documentation>**Identification of the schema version.**</documentation>  </annotation>  </attribute>  </complexType>  </element>  <element name=**"RegisterCashDepositResponse"**>  <annotation>  <documentation>**Root XML element representing cash deposit response message.**</documentation>  </annotation>  <complexType>  <all minOccurs=**"1"** maxOccurs=**"1"**>  <element name=**"Header"** type=**"me:RegisterCashDepositResponseHeaderType"** minOccurs=**"1"** maxOccurs=**"1"**>  <annotation>  <documentation>**Element representing the header of the message.**</documentation>  </annotation>  </element>  <element name=**"FCDC"** type=**"me:UUIDSType"** minOccurs=**"1"** maxOccurs=**"1"**>  <annotation>  <documentation>**Fiscalization cash deposit code generated by the CIS.**</documentation>  </annotation>  </element>  <element ref=**"ds:Signature"** minOccurs=**"1"** maxOccurs=**"1"**/>  </all>  <attribute name=**"Id"** type=**"string"** use=**"required"** fixed=**"Response"**>  <annotation>  <documentation>**Identification of the response, used to reference a signature.**</documentation>  </annotation>  </attribute>  <attribute name=**"Version"** type=**"me:IntSType"** use=**"required"** fixed=**"1"**>  <annotation>  <documentation>**Identification of the schema version.**</documentation>  </annotation>  </attribute>  </complexType>  </element>  <element name=**"RegisterTCRRequest"**>  <annotation>  <documentation>**Root XML element representing TCR request message.**</documentation>  </annotation>  <complexType>  <all minOccurs=**"1"** maxOccurs=**"1"**>  <element name=**"Header"** type=**"me:RegisterTCRRequestHeaderType"** minOccurs=**"1"** maxOccurs=**"1"**>  <annotation>  <documentation>**Element representing the header of the message.**</documentation>  </annotation>  </element>  <element name=**"TCR"** type=**"me:TCRType"** minOccurs=**"1"** maxOccurs=**"1"**>  <annotation>  <documentation>**XML element representing a single TCR registration message.**</documentation>  </annotation>  </element>  <element ref=**"ds:Signature"** minOccurs=**"1"** maxOccurs=**"1"**/>  </all>  <attribute name=**"Id"** type=**"string"** use=**"required"** fixed=**"Request"**>  <annotation>  <documentation>**Identification of the request, used to reference a signature.**</documentation>  </annotation>  </attribute>  <attribute name=**"Version"** type=**"me:IntSType"** use=**"required"** fixed=**"1"**>  <annotation>  <documentation>**Identification of the schema version.**</documentation>  </annotation>  </attribute>  </complexType>  </element>  <element name=**"RegisterTCRResponse"**>  <annotation>  <documentation>**Root XML element representing TCR response message.**</documentation>  </annotation>  <complexType>  <all minOccurs=**"1"** maxOccurs=**"1"**>  <element name=**"Header"** type=**"me:RegisterTCRResponseHeaderType"** minOccurs=**"1"** maxOccurs=**"1"**>  <annotation>  <documentation>**Element representing the header of the message.**</documentation>  </annotation>  </element>  <element name=**"TCRCode"** type=**"me:RegistrationCodeSType"** minOccurs=**"1"** maxOccurs=**"1"**>  <annotation>  <documentation>**TCR code generated by the service.**</documentation>  </annotation>  </element>  <element ref=**"ds:Signature"** minOccurs=**"1"** maxOccurs=**"1"**/>  </all>  <attribute name=**"Id"** type=**"string"** use=**"required"** fixed=**"Response"**>  <annotation>  <documentation>**Identification of the response, used to reference a signature.**</documentation>  </annotation>  </attribute>  <attribute name=**"Version"** type=**"me:IntSType"** use=**"required"** fixed=**"1"**>  <annotation>  <documentation>**Identification of the schema version.**</documentation>  </annotation>  </attribute>  </complexType>  </element>    <complexType name=**"RegisterInvoiceRequestHeaderType"**>  <attribute name=**"UUID"** type=**"me:UUIDSType"** use=**"required"**>  <annotation>  <documentation>**UUID generated by a TCR for every register sale data message send to the CIS.**</documentation>  </annotation>  </attribute>  <attribute name=**"SendDateTime"** type=**"me:UTCSType"** use=**"required"**>  <annotation>  <documentation>**Date and time of sending the register invoice data message from a TCR to the CIS.**</documentation>  </annotation>  </attribute>  <attribute name=**"SubseqDelivType"** type=**"me:SubseqDelivTypeSType"** use=**"optional"** >  <annotation>  <documentation>**Type of subsequent delivery if message is delivered after invoice issuance.**</documentation>  </annotation>  </attribute>  </complexType>  <complexType name=**"RegisterInvoiceResponseHeaderType"**>  <attribute name=**"UUID"** type=**"me:UUIDSType"** use=**"required"**>  <annotation>  <documentation>**Element generated by the CIS for every message sent to the TCR. It uniquely identifies the message sent to the TCR.**</documentation>  </annotation>  </attribute>  <attribute name=**"RequestUUID"** type=**"me:UUIDSType"** use=**"required"**>  <annotation>  <documentation>**Element generated by the TCR and referenced by the CIS. It uniquely identifies the request message for which response message was sent to the TCR.**</documentation>  </annotation>  </attribute>  <attribute name=**"SendDateTime"** type=**"me:UTCSType"** use=**"required"**>  <annotation>  <documentation>**Element represents date and time of sending the response message to the TCR.** </documentation>  </annotation>  </attribute>  </complexType>  <complexType name=**"RegisterCashDepositRequestHeaderType"**>  <attribute name=**"UUID"** type=**"me:UUIDSType"** use=**"required"** >  <annotation>  <documentation>**Element generated by the TCR. It uniquely identifies the request message sent from TCR to CIS.**</documentation>  </annotation>  </attribute>  <attribute name=**"SendDateTime"** type=**"me:UTCSType"** use=**"required"** >  <annotation>  <documentation>**Element represents date and time of sending the request message to the CIS.**</documentation>  </annotation>  </attribute>  <attribute name=**"SubseqDelivType"** type=**"me:SubseqDelivTypeSType"** use=**"optional"** >  <annotation>  <documentation>**Type of subsequent delivery if message is delivered after cash deposit issuance.**</documentation>  </annotation>  </attribute>  </complexType>  <complexType name=**"RegisterCashDepositResponseHeaderType"**>  <attribute name=**"UUID"** type=**"me:UUIDSType"** use=**"required"**>  <annotation>  <documentation>**Element generated by the CIS. It uniquely identifies the response message sent from CIS to TCR.**</documentation>  </annotation>  </attribute>  <attribute name=**"RequestUUID"** type=**"me:UUIDSType"** use=**"required"**>  <annotation>  <documentation>**Element generated by the TCR and referenced by the CIS. It uniquely identifies the request message for which response message was sent to the TCR.**</documentation>  </annotation>  </attribute>  <attribute name=**"SendDateTime"** type=**"me:UTCSType"** use=**"required"**>  <annotation>  <documentation>**Element represents date and time of sending the response message to the TCR.** </documentation>  </annotation>  </attribute>  </complexType>  <complexType name=**"RegisterTCRRequestHeaderType"**>  <attribute name=**"UUID"** type=**"me:UUIDSType"** use=**"required"**>  <annotation>  <documentation>**Element generated by the TCR. It uniquely identifies the request message sent from TCR to CIS.**</documentation>  </annotation>  </attribute>  <attribute name=**"SendDateTime"** type=**"me:UTCSType"** use=**"required"**>  <annotation>  <documentation>**Element represents date and time of sending the request message to the CIS.**</documentation>  </annotation>  </attribute>  </complexType>  <complexType name=**"RegisterTCRResponseHeaderType"**>  <attribute name=**"UUID"** type=**"me:UUIDSType"** use=**"required"**>  <annotation>  <documentation>**Element generated by the CIS. It uniquely identifies the response message sent from CIS to TCR.** </documentation>  </annotation>  </attribute>  <attribute name=**"RequestUUID"** type=**"me:UUIDSType"** use=**"required"**>  <annotation>  <documentation>**Element generated by the TCR and referenced by the CIS. It uniquely identifies the request message for which the response message was sent to the TCR.**</documentation>  </annotation>  </attribute>  <attribute name=**"SendDateTime"** type=**"me:UTCSType"** use=**"required"**>  <annotation>  <documentation>**Element represents date and time of sending the response message to the TCR.** </documentation>  </annotation>  </attribute>  </complexType>  <complexType name=**"InvoiceType"**>  <all>  <element name=**"SupplyDateOrPeriod"** type=**"me:SupplyDateOrPeriodType"** minOccurs=**"0"** maxOccurs=**"1"**>  <annotation>  <documentation>**XML element representing supply date or period is supply is different from the date when the invoice was issued.**</documentation>  </annotation>  </element>  <element name=**"CorrectiveInv"** type=**"me:CorrectiveInvType"** minOccurs=**"0"** maxOccurs=**"1"**>  <annotation>  <documentation>**XML element groups data for a corrective invoice.**</documentation>  </annotation>  </element>  <element name=**"PayMethods"** type=**"me:PayMethodsType"** minOccurs=**"1"** maxOccurs=**"1"**>  <annotation>  <documentation>**XML element representing list of payment methods.**</documentation>  </annotation>  <unique name=**"UniquePayMethodType"**>  <selector xpath=**".//me:PayMethod"**/>  <field xpath=**"@Type"**/>  </unique>  </element>  <element name=**"Currency"** type=**"me:CurrencyType"** minOccurs=**"0"** maxOccurs=**"1"**>  <annotation>  <documentation>**XML element representing currency in which the amount on the invoice is expressed.**</documentation>  </annotation>  </element>  <element name=**"Seller"** type=**"me:SellerType"** minOccurs=**"1"** maxOccurs=**"1"** >  <annotation>  <documentation>**XML element representing an seller of the goods.**</documentation>  </annotation>  </element>  <element name=**"Buyer"** type=**"me:BuyerType"** minOccurs=**"0"** maxOccurs=**"1"** >  <annotation>  <documentation>**XML element representing a buyer of the goods.**</documentation>  </annotation>  </element>  <element name=**"Items"** type=**"me:InvoiceItemsType"** minOccurs=**"1"** maxOccurs=**"1"** >  <annotation>  <documentation>**XML element representing list of invoice items (goods or services).**</documentation>  </annotation>  </element>  <element name=**"SameTaxes"** type=**"me:SameTaxesType"** minOccurs=**"0"** maxOccurs=**"1"** >  <annotation>  <documentation>**XML element representing list of items of the same VAT rate.**</documentation>  </annotation>  </element>  <element name=**"Fees"** type=**"me:FeesType"** minOccurs=**"0"** maxOccurs=**"1"** >  <annotation>  <documentation>**XML element representing list of fees.**</documentation>  </annotation>  <unique name=**"UniqueFeesType"**>  <selector xpath=**".//me:Fee"**/>  <field xpath=**"@Type"**/>  </unique>  </element>  <element name=**"SumInvIICRefs"** type=**"me:SumInvIICRefsType"** minOccurs=**"0"** maxOccurs=**"1"** >  <annotation>  <documentation>**XML element representing list of invoices IIC referenced by a summary invoice.**</documentation>  </annotation>  </element>  <element name=**"BadDebtInv"** type=**"me:BadDebtInvType"** minOccurs=**"0"** maxOccurs=**"1"**>  <annotation>  <documentation>**XML element groups data for an original invoice that will be declared bad debt invoice, as uncollectible.**</documentation>  </annotation>  </element>  </all>  <attribute name=**"TypeOfInv"** type=**"me:InvoiceSType"** use=**"required"** >  <annotation>  <documentation>**Type of the item represents the type of invoice item, e.g. regular sale or a returned item.**</documentation>  </annotation>  </attribute>  <attribute name=**"IsSimplifiedInv"** type=**"boolean"** use=**"required"** >  <annotation>  <documentation>**If the invoice is simplified or not.**</documentation>  </annotation>  </attribute>  <attribute name=**"TypeOfSelfIss"** type=**"me:SelfIssSType"** use=**"optional"** >  <annotation>  <documentation>**Type of self-issuing.**</documentation>  </annotation>  </attribute>  <attribute name=**"IssueDateTime"** type=**"me:UTCSType"** use=**"required"** >  <annotation>  <documentation>**Date and time of invoice created and issued at TCR.**</documentation>  </annotation>  </attribute>  <attribute name=**"InvNum"** type=**"me:InvNumSType"** use=**"required"** >  <annotation>  <documentation>**Invoice number composed of invoice ordinal number, year of invoice issuance and code of TCR that issued invoice.**</documentation>  </annotation>  </attribute>  <attribute name=**"InvOrdNum"** type=**"me:IntSType"** use=**"required"** >  <annotation>  <documentation>**Invoice ordinal number.**</documentation>  </annotation>  </attribute>  <attribute name=**"TCRCode"** type=**"me:RegistrationCodeSType"** use=**"required"** >  <annotation>  <documentation>**Code of the device that issued the invoice.**</documentation>  </annotation>  </attribute>  <attribute name=**"IsIssuerInVAT"** type=**"boolean"** use=**"required"** >  <annotation>  <documentation>**Is the taxpayer in the VAT system.**</documentation>  </annotation>  </attribute>  <attribute name=**"TaxFreeAmt"** type=**"me:DecimalNegSType"** use=**"optional"** >  <annotation>  <documentation>**Amount on items that are tax free.**</documentation>  </annotation>  </attribute>  <attribute name=**"MarkUpAmt"** type=**"me:DecimalNegSType"** use=**"optional"** >  <annotation>  <documentation>**Amount of the mark-up on the invoice.**</documentation>  </annotation>  </attribute>  <attribute name=**"GoodsExAmt"** type=**"me:DecimalNegSType"** use=**"optional"** >  <annotation>  <documentation>**Total price of delivery of exported goods. There is no VAT on the invoice.**</documentation>  </annotation>  </attribute>  <attribute name=**"TotPriceWoVAT"** type=**"me:DecimalNegSType"** use=**"required"** >  <annotation>  <documentation>**Total price of the invoice without VAT.**</documentation>  </annotation>  </attribute>  <attribute name=**"TotVATAmt"** type=**"me:DecimalNegSType"** use=**"optional"** >  <annotation>  <documentation>**Total VAT amount.**</documentation>  </annotation>  </attribute>  <attribute name=**"TotPrice"** type=**"me:DecimalNegSType"** use=**"required"** >  <annotation>  <documentation>**Total price of the invoice including VAT.**</documentation>  </annotation>  </attribute>  <attribute name=**"OperatorCode"** type=**"me:RegistrationCodeSType"** use=**"required"** >  <annotation>  <documentation>**Reference to the operator who is operating on TCR. Value represents code of the operator.**</documentation>  </annotation>  </attribute>  <attribute name=**"BusinUnitCode"** type=**"me:RegistrationCodeSType"** use=**"required"** >  <annotation>  <documentation>**Code (ID) of the business unit in which the invoice is issued.**</documentation>  </annotation>  </attribute>  <attribute name=**"SoftCode"** type=**"me:RegistrationCodeSType"** use=**"required"** >  <annotation>  <documentation>**Number of the software used for invoice issuing.**</documentation>  </annotation>  </attribute>  <attribute name=**"IIC"** type=**"me:Hex32SType"** use=**"required"** >  <annotation>  <documentation>**Issuers’s invoice code which is generated by the cash register of the issuer of the invoice. This is a unique code for every invoice.** </documentation>  </annotation>  </attribute>  <attribute name=**"IICSignature"** type=**"me:Hex512SType"** use=**"required"** >  <annotation>  <documentation>**Signed issuer’s invoice code concatenated parameters.**</documentation>  </annotation>  </attribute>  <attribute name=**"IsReverseCharge"** type=**"boolean"** use=**"required"** >  <annotation>  <documentation>**Buyer is obliged to pay taxes by himself rather than issuer does it for him.**</documentation>  </annotation>  </attribute>  <attribute name=**"PayDeadline"** type=**"me:DateSType"** use=**"optional"** >  <annotation>  <documentation>**Last day for a payment.**</documentation>  </annotation>  </attribute>  <attribute name=**"ParagonBlockNum"** type=**"me:String20SType"** use=**"optional"** >  <annotation>  <documentation>**Number of paragon block.**</documentation>  </annotation>  </attribute>  <attribute name="TaxPeriod" type="me:TaxPeriodSType" use="optional" >  <annotation>  <documentation>Tax period for tax return.</documentation>  </annotation>  </attribute>  </complexType>  <complexType name=**"InvoiceItemType"**>  <all>  <element name=**"VS"** type=**"me:VouchersSoldType"** minOccurs=**"0"** maxOccurs=**"1"**>  <annotation>  <documentation>**Element representing voucher data linked to this item.**</documentation>  </annotation>  </element>  </all>  <attribute name=**"N"** type=**"me:String50SType"** use=**"required"**>  <annotation>  <documentation>**Name of the item.**</documentation>  </annotation>  </attribute>  <attribute name=**"C"** type=**"me:String50SType"** use=**"optional"**>  <annotation>  <documentation>**Code of the item.**</documentation>  </annotation>  </attribute>  <attribute name=**"U"** type=**"me:String50SType"** use=**"required"**>  <annotation>  <documentation>**Unit of measure for specific item – piece, weight, length...**</documentation>  </annotation>  </attribute>  <attribute name=**"Q"** type=**"me:DoubleNegForQuantitySType"** use=**"required"**>  <annotation>  <documentation>**Amount or number (quantity) of items.**</documentation>  </annotation>  </attribute>  <attribute name=**"UPB"** type=**"me:DecimalSType"** use=**"required"**>  <annotation>  <documentation>**Price of one item before VAT is applied (unit price without VAT).**</documentation>  </annotation>  </attribute>  <attribute name=**"UPA"** type=**"me:DecimalSType"** use=**"required"**>  <annotation>  <documentation>**Price of one item after VAT is applied (unit price with VAT). It is calculated as PA/Q.**</documentation>  </annotation>  </attribute>  <attribute name=**"R"** type=**"me:DecimalSType"** use=**"optional"**>  <annotation>  <documentation>**Rebate percentage.**</documentation>  </annotation>  </attribute>  <attribute name=**"RR"** type=**"boolean"** use=**"optional"**>  <annotation>  <documentation>**Is rebate reducing base price?**</documentation>  </annotation>  </attribute>  <attribute name=**"PB"** type=**"me:DecimalNegSType"** use=**"required"**>  <annotation>  <documentation>**Price before VAT for the items in this group of items. This is not the unit price of the item. It is the unit price multiplied by the quantity of items.**</documentation>  </annotation>  </attribute>  <attribute name=**"VR"** type=**"me:DecimalSType"** use=**"optional"**>  <annotation>  <documentation>**Rate of value added tax expressed as percentage.**</documentation>  </annotation>  </attribute>  <attribute name=**"VA"** type=**"me:DecimalNegSType"** use=**"optional"**>  <annotation>  <documentation>**Amount of value added tax for the items in this group of items.**</documentation>  </annotation>  </attribute>  <attribute name=**"IN"** type=**"boolean"** use=**"optional"** >  <annotation>  <documentation>**If true, the item is investment.**</documentation>  </annotation>  </attribute>  <attribute name=**"PA"** type=**"me:DecimalNegSType"** use=**"required"**>  <annotation>  <documentation>**Price after applying VAT for the items in this group of items.**</documentation>  </annotation>  </attribute>  <attribute name=**"EX"** type=**"me:ExemptFromVATSType"** use=**"optional"**>  <annotation>  <documentation>**Exempt from VAT.**</documentation>  </annotation>  </attribute>  </complexType>  <complexType name=**"InvoiceItemsType"**>  <sequence>  <element name=**"I"** type=**"me:InvoiceItemType"** minOccurs=**"1"** maxOccurs=**"1000"**>  <annotation>  <documentation>**Element representing a single item on the invoice.**</documentation>  </annotation>  </element>  </sequence>  </complexType>  <complexType name=**"SupplyDateOrPeriodType"**>  <attribute name=**"Start"** type=**"me:DateSType"** use=**"required"**>  <annotation>  <documentation>**Start date.**</documentation>  </annotation>  </attribute>  <attribute name=**"End"** type=**"me:DateSType"** use=**"required"**>  <annotation>  <documentation>**End date.**</documentation>  </annotation>  </attribute>  </complexType>  <complexType name=**"CorrectiveInvType"**>  <attribute name=**"IICRef"** type=**"me:Hex32SType"** use=**"required"**>  <annotation>  <documentation>**IIC reference on the original invoice.**</documentation>  </annotation>  </attribute>  <attribute name=**"IssueDateTime"** type=**"me:UTCSType"** use=**"required"**>  <annotation>  <documentation>**Date and time the original invoice is created and issued at TCR**</documentation>  </annotation>  </attribute>  <attribute name=**"Type"** type=**"me:CorrectiveInvTypeSType"** use=**"required"**>  <annotation>  <documentation>**Type of the corrective invoice.**</documentation>  </annotation>  </attribute>  </complexType>  <complexType name=**"CurrencyType"**>  <attribute name=**"Code"** type=**"me:CurrencyCodeSType"** use=**"required"**>  <annotation>  <documentation>**Currency code from the ISO 4217 standard.**</documentation>  </annotation>  </attribute>  <attribute name=**"ExRate"** type=**"me:DoubleSType"** use=**"required"**>  <annotation>  <documentation>**Exchange rate applied to calculate the equivalent amount of foreign currency for the total amount expressed in LEK. Exchange rate express equivalent amount of LEK for 1 unit of foreign currency.**</documentation>  </annotation>  </attribute>  </complexType>  <complexType name=**"SellerType"**>  <attribute name=**"IDType"** type=**"me:IDTypeSType"** use=**"required"**>  <annotation>  <documentation>**Seller's identification number type.**</documentation>  </annotation>  </attribute>  <attribute name=**"IDNum"** type=**"me:String20SType"** use=**"required"**>  <annotation>  <documentation>**Seller's identification number.**</documentation>  </annotation>  </attribute>  <attribute name=**"Name"** type=**"me:String100SType"** use=**"required"**>  <annotation>  <documentation>**Seller's name.**</documentation>  </annotation>  </attribute>  <attribute name=**"Address"** type=**"me:String200SType"** use=**"optional"**>  <annotation>  <documentation>**Seller's address.**</documentation>  </annotation>  </attribute>  <attribute name=**"Town"** type=**"me:String100SType"** use=**"optional"**>  <annotation>  <documentation>**Seller's town.**</documentation>  </annotation>  </attribute>  <attribute name=**"Country"** type=**"me:CountryCodeSType"** use=**"optional"**>  <annotation>  <documentation>**Seller's country.**</documentation>  </annotation>  </attribute>  </complexType>  <complexType name=**"BuyerType"**>  <attribute name=**"IDType"** type=**"me:IDTypeSType"** use=**"optional"**>  <annotation>  <documentation>**Buyer's identification number type.**</documentation>  </annotation>  </attribute>  <attribute name=**"IDNum"** type=**"me:String20SType"** use=**"optional"**>  <annotation>  <documentation>**Buyer's identification number.**</documentation>  </annotation>  </attribute>  <attribute name=**"Name"** type=**"me:String100SType"** use=**"optional"**>  <annotation>  <documentation>**Buyer's name.**</documentation>  </annotation>  </attribute>  <attribute name=**"Address"** type=**"me:String200SType"** use=**"optional"**>  <annotation>  <documentation>**Buyer's address.**</documentation>  </annotation>  </attribute>  <attribute name=**"Town"** type=**"me:String100SType"** use=**"optional"**>  <annotation>  <documentation>**Buyer's town.**</documentation>  </annotation>  </attribute>  <attribute name=**"Country"** type=**"me:CountryCodeSType"** use=**"optional"**>  <annotation>  <documentation>**Buyer's country.**</documentation>  </annotation>  </attribute>  </complexType>  <complexType name=**"PayMethodsType"**>  <sequence>  <element name=**"PayMethod"** type=**"me:PayMethodType"** minOccurs=**"1"** maxOccurs=**"10"**>  <annotation>  <documentation>**XML element representing one payment method.**</documentation>  </annotation>  </element>  </sequence>  </complexType>  <complexType name=**"SameTaxesType"**>  <sequence>  <element name=**"SameTax"** type=**"me:SameTaxType"** minOccurs=**"1"** maxOccurs=**"20"**>  <annotation>  <documentation>**Element representing an item of list of same tax items.**</documentation>  </annotation>  </element>  </sequence>  </complexType>  <complexType name=**"FeesType"**>  <sequence>  <element name=**"Fee"** type=**"me:FeeType"** minOccurs=**"1"** maxOccurs=**"20"**>  <annotation>  <documentation>**Element representing an item of list of fee items.**</documentation>  </annotation>  </element>  </sequence>  </complexType>  <complexType name=**"SumInvIICRefsType"**>  <sequence>  <element name=**"SumInvIICRef"** type=**"me:SumInvIICRefType"** minOccurs=**"1"** maxOccurs=**"1000"**>  <annotation>  <documentation>**XML element representing a single IIC reference on the list of SumInvIICRefs.**</documentation>  </annotation>  </element>  </sequence>  </complexType>  <complexType name=**"PayMethodType"**>  <sequence>  <element name=**"Vouchers"** type=**"me:VouchersType"** minOccurs=**"0"** maxOccurs=**"1"**>  <annotation>  <documentation>**XML element representing list of voucher numbers if the payment method is voucher.**</documentation>  </annotation>  </element>  </sequence>  <attribute name=**"Type"** type=**"me:PaymentMethodTypeSType"** use=**"required"**>  <annotation>  <documentation>**Type of the payment method.**</documentation>  </annotation>  </attribute>  <attribute name=**"Amt"** type=**"me:DecimalNegSType"** use=**"required"**>  <annotation>  <documentation>**Amount payed by payment method in the country currency.**</documentation>  </annotation>  </attribute>  <attribute name=**"CompCard"** type=**"me:String50SType"** use=**"optional"**>  <annotation>  <documentation>**Company card number if the payment method is company cards.**</documentation>  </annotation>  </attribute>  </complexType>  <complexType name=**"VouchersType"**>  <sequence>  <element name=**"Voucher"** type=**"me:VoucherType"** minOccurs=**"1"** maxOccurs=**"20"**>  <annotation>  <documentation>**XML element representing a single voucher number.**</documentation>  </annotation>  </element>  </sequence>  </complexType>  <complexType name=**"VoucherType"**>  <attribute name=**"Num"** type=**"me:VoucherNumSType"** use=**"required"**>  <annotation>  <documentation>**XML element representing single voucher serial number.**</documentation>  </annotation>  </attribute>  </complexType>  <complexType name=**"SameTaxType"**>  <attribute name=**"NumOfItems"** type=**"me:IntSType"** use=**"required"**>  <annotation>  <documentation>**Number of items.**</documentation>  </annotation>  </attribute>  <attribute name=**"PriceBefVAT"** type=**"me:DecimalNegSType"** use=**"required"**>  <annotation>  <documentation>**Price before VAT.**</documentation>  </annotation>  </attribute>  <attribute name=**"VATRate"** type=**"me:DecimalNegSType"** use=**"optional"**>  <annotation>  <documentation>**VAT rate.**</documentation>  </annotation>  </attribute>  <attribute name=**"ExemptFromVAT"** type=**"me:ExemptFromVATSameTaxItemSType"** use=**"optional"**>  <annotation>  <documentation>**Exempt from VAT.**</documentation>  </annotation>  </attribute>  <attribute name=**"VATAmt"** type=**"me:DecimalNegSType"** use=**"optional"**>  <annotation>  <documentation>**VAT amount.**</documentation>  </annotation>  </attribute>  </complexType>  <complexType name=**"FeeType"**>  <attribute name=**"Type"** type=**"me:FeeTypeSType"** use=**"required"**>  <annotation>  <documentation>**Type of the fee.**</documentation>  </annotation>  </attribute>  <attribute name=**"Amt"** type=**"me:DecimalNegSType"** use=**"required"**>  <annotation>  <documentation>**The decimal amount of the fee in the country currency.**</documentation>  </annotation>  </attribute>  </complexType>  <complexType name=**"SumInvIICRefType"**>  <attribute name=**"IIC"** type=**"me:Hex32SType"** use=**"required"**>  <annotation>  <documentation>**IIC of the invoice that is referenced by the summary invoice.**</documentation>  </annotation>  </attribute>  <attribute name=**"IssueDateTime"** type=**"me:UTCSType"** use=**"required"**>  <annotation>  <documentation>**Date and time the invoice referenced by the summary invoice is created and issued at TCR.**</documentation>  </annotation>  </attribute>  </complexType>  <complexType name=**"CashDepositType"**>  <attribute name=**"ChangeDateTime"** type=**"me:UTCSType"** use=**"required"** >  <annotation>  <documentation>**Element representing date and time when the deposit of cash was checked in the cash register.**</documentation>  </annotation>  </attribute>  <attribute name=**"Operation"** type=**"me:CashDepositOperationSType"** use=**"required"**>  <annotation>  <documentation>**This is an XML element that represents the operation made at the cash register.**</documentation>  </annotation>  </attribute>  <attribute name=**"CashAmt"** type=**"me:DecimalSType"** use=**"required"** >  <annotation>  <documentation>**Element representing the amount of cash found in the cash register after the operation.**</documentation>  </annotation>  </attribute>  <attribute name=**"TCRCode"** type=**"me:RegistrationCodeSType"** use=**"required"** >  <annotation>  <documentation>**Element representing the unique code of the TCR in question.**</documentation>  </annotation>  </attribute>  <attribute name=**"IssuerTIN"** type=**"me:TINSType"** use=**"required"**>  <annotation>  <documentation>**Element representing issuer's TIN (tax number).**</documentation>  </annotation>  </attribute>  </complexType>  <complexType name=**"TCRType"**>  <simpleContent>  <extension base=**"string"**>  <attribute name=**"IssuerTIN"** type=**"me:TINSType"** use=**"required"**>  <annotation>  <documentation>**Issuer's PIB/JMB.**</documentation>  </annotation>  </attribute>  <attribute name=**"BusinUnitCode"** type=**"me:RegistrationCodeSType"** use=**"required"**>  <annotation>  <documentation>**Code of the business unit in which the invoice is issued.**</documentation>  </annotation>  </attribute>  <attribute name=**"TCRIntID"** type=**"me:String50SType"** use=**"required"**>  <annotation>  <documentation>**Element representing the internal identification of the TCR.**</documentation>  </annotation>  </attribute>  <attribute name=**"SoftCode"** type=**"me:RegistrationCodeSType"** use=**"optional"**>  <annotation>  <documentation>**Code of the software used for invoice issuing.**</documentation>  </annotation>  </attribute>  <attribute name=**"MaintainerCode"** type=**"me:RegistrationCodeSType"** use=**"optional"**>  <annotation>  <documentation>**Code of the maintainer of the software used for invoice issuing.**</documentation>  </annotation>  </attribute>  <attribute name=**"ValidFrom"** type=**"me:DateSType"** use=**"optional"**>  <annotation>  <documentation>**Date from which the TCR will be used.**</documentation>  </annotation>  </attribute>  <attribute name=**"ValidTo"** type=**"me:DateSType"** use=**"optional"**>  <annotation>  <documentation>**Date until the TCR will be used.**</documentation>  </annotation>  </attribute>  <attribute name=**"Type"** type=**"me:TCRSType"** use=**"optional"**>  <annotation>  <documentation>**TCR type.**</documentation>  </annotation>  </attribute>  </extension>  </simpleContent>  </complexType>  <complexType name=**"VouchersSoldType"**>  <all>  <element name=**"VD"** type=**"me:VoucherSoldDataType"** minOccurs=**"1"** maxOccurs=**"1"**>  <annotation>  <documentation>**XML element representing a voucher data.**</documentation>  </annotation>  </element>  <element name=**"VN"** type=**"me:VouchersSoldNumType"** minOccurs=**"1"** maxOccurs=**"1"**>  <annotation>  <documentation>**XML element representing a voucher numbers.**</documentation>  </annotation>  </element>  </all>  </complexType>  <complexType name=**"VoucherSoldDataType"**>  <attribute name=**"D"** type=**"me:DateSType"** use=**"required"**>  <annotation>  <documentation>**Expiration date of the voucher.**</documentation>  </annotation>  </attribute>  <attribute name=**"N"** type=**"me:DecimalSType"** use=**"required"**>  <annotation>  <documentation>**Nominal voucher value.**</documentation>  </annotation>  </attribute>  </complexType>  <complexType name=**"VouchersSoldNumType"**>  <sequence>  <element name=**"V"** type=**"me:VoucherType"** minOccurs=**"1"** maxOccurs=**"1000"**>  <annotation>  <documentation>**XML element representing a single voucher number sold.**</documentation>  </annotation>  </element>  </sequence>  </complexType>  <complexType name=**"BadDebtInvType"**>  <attribute name=**"IICRef"** type=**"me:Hex32SType"** use=**"required"**>  <annotation>  <documentation>**IIC reference on the original invoice.**</documentation>  </annotation>  </attribute>  <attribute name=**"IssueDateTime"** type=**"me:UTCSType"** use=**"required"**>  <annotation>  <documentation>**Date and time the original invoice is created and issued at TCR**</documentation>  </annotation>  </attribute>  </complexType>    <simpleType name=**"InvoiceSType"**>  <annotation>  <documentation>**Invoice type.**</documentation>  </annotation>  <restriction base=**"string"**>  <enumeration value=**"CASH"**>  <annotation>  <documentation>**Cash invoice.**</documentation>  </annotation>  </enumeration>  <enumeration value=**"NONCASH"**>  <annotation>  <documentation>**Non-cash invoice.**</documentation>  </annotation>  </enumeration>  </restriction>  </simpleType>  <simpleType name=**"TCRSType"**>  <annotation>  <documentation>**TCR type.**</documentation>  </annotation>  <restriction base=**"string"**>  <enumeration value=**"REGULAR"**>  <annotation>  <documentation>**Regular TCR.**</documentation>  </annotation>  </enumeration>  <enumeration value=**"VENDING"**>  <annotation>  <documentation>**Self-vending machine.**</documentation>  </annotation>  </enumeration>  </restriction>  </simpleType>  <simpleType name=**"DecimalSType"**>  <annotation>  <documentation>**Decimal number with two numbers after decimal point.**</documentation>  </annotation>  <restriction base=**"decimal"**>  <pattern value=**"([1-9][0-9]\*|0)\.[0-9]{2}|0"**/>  </restriction>  </simpleType>  <simpleType name=**"SelfIssSType"**>  <annotation>  <documentation>**Self issuing type.**</documentation>  </annotation>  <restriction base=**"string"**>  <enumeration value=**"AGREEMENT"**>  <annotation>  <documentation>**The previous agreement between the parties.**</documentation>  </annotation>  </enumeration>  <enumeration value=**"DOMESTIC"**>  <annotation>  <documentation>**Buying from domestic farmers.**</documentation>  </annotation>  </enumeration>  <enumeration value=**"ABROAD"**>  <annotation>  <documentation>**Buying services from abroad.**</documentation>  </annotation>  </enumeration>  <enumeration value=**"SELF"**>  <annotation>  <documentation>**Self-consumption.**</documentation>  </annotation>  </enumeration>  <enumeration value=**"OTHER"**>  <annotation>  <documentation>**Other**</documentation>  </annotation>  </enumeration>  </restriction>  </simpleType>  <simpleType name=**"PaymentMethodTypeSType"**>  <annotation>  <documentation>**Payment method type.**</documentation>  </annotation>  <restriction base=**"string"**>  <enumeration value=**"BANKNOTE"**>  <annotation>  <documentation>**Banknotes and coins**</documentation>  </annotation>  </enumeration>  <enumeration value=**"CARD"**>  <annotation>  <documentation>**Credit and debit card issued to the physical person**</documentation>  </annotation>  </enumeration>  <enumeration value=**"BUSINESSCARD"**>  <annotation>  <documentation>**Credit and debit card issued to the taxpayer**</documentation>  </annotation>  </enumeration>  <enumeration value=**"SVOUCHER"**>  <annotation>  <documentation>**Single-purpose voucher**</documentation>  </annotation>  </enumeration>  <enumeration value=**"COMPANY"**>  <annotation>  <documentation>**Seller's company cards, gift cards and similar prepaid cards**</documentation>  </annotation>  </enumeration>  <enumeration value=**"ORDER"**>  <annotation>  <documentation>**Invoice not yet paid. It will be paid by summary invoice.**</documentation>  </annotation>  </enumeration>  <enumeration value=**"ADVANCE"**>  <annotation>  <documentation>**Invoice is paid in advance**</documentation>  </annotation>  </enumeration>  <enumeration value=**"ACCOUNT"**>  <annotation>  <documentation>**Transaction account**</documentation>  </annotation>  </enumeration>  <enumeration value=**"FACTORING"**>  <annotation>  <documentation>**Factoring**</documentation>  </annotation>  </enumeration>  <enumeration value=**"OTHER"**>  <annotation>  <documentation>**Other cashless payments**</documentation>  </annotation>  </enumeration>  <enumeration value=**"OTHER-CASH"**>  <annotation>  <documentation>**Other cash payments**</documentation>  </annotation>  </enumeration>  </restriction>  </simpleType>  <simpleType name=**"UTCSType"**>  <annotation>  <documentation>**Date and time represented as UTC time with ISO 8601 format.**</documentation>  </annotation>  <restriction base=**"dateTime"**>  <pattern value=**"[0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}[+-][0-9]{2}:[0-9]{2}"**/>  </restriction>  </simpleType>  <simpleType name=**"DateSType"**>  <annotation>  <documentation>**Date expressed as year-month-day.**</documentation>  </annotation>  <restriction base=**"date"**>  <pattern value=**"[0-9]{4}-[0-9]{2}-[0-9]{2}"**/>  </restriction>  </simpleType>  <simpleType name=**"UUIDSType"**>  <annotation>  <documentation>**UUID constructed according to the RFC4122 (https://tools.ietf.org/html/rfc4122).**</documentation>  </annotation>  <restriction base=**"string"**>  <pattern value=**"[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[1-5][0-9a-fA-F]{3}-[89abAB][0-9a-fA-F]{3}-[0-9a-fA-F]{12}"**/>  </restriction>  </simpleType>  <simpleType name=**"VoucherNumSType"**>  <annotation>  <documentation>**Voucher serial number.**</documentation>  </annotation>  <restriction base=**"string"**>  <pattern value=**"[1-9]{1}[0-9]{0,7}-[0-9]{4}-[a-zA-Z]{1}[0-9]{8}[a-zA-Z]{1}"**/>  </restriction>  </simpleType>  <simpleType name=**"TINSType"**>  <annotation>  <documentation>**PIB or JMB numebr.**</documentation>  </annotation>  <restriction base=**"string"**>  <pattern value=**"[0-9]{13}|[0-9]{8}"**/>  </restriction>  </simpleType>  <simpleType name=**"InvNumSType"**>  <annotation>  <documentation>**Invoice number composed of Invoice ordinal number, year of invoice issuing and TCR code.**</documentation>  </annotation>  <restriction base=**"string"**>  <pattern value=**"[a-z]{2}[0-9]{3}[a-z]{2}[0-9]{3}\/[1-9]{1}[0-9]{0,14}\/[0-9]{4}\/[a-z]{2}[0-9]{3}[a-z]{2}[0-9]{3}"**/>  </restriction>  </simpleType>  <simpleType name=**"String200SType"**>  <annotation>  <documentation>**String of up to 200 characters.**</documentation>  </annotation>  <restriction base=**"string"**>  <minLength value=**"1"**/>  <maxLength value=**"200"**/>  </restriction>  </simpleType>  <simpleType name=**"String100SType"**>  <annotation>  <documentation>**String of up to 100 characters.**</documentation>  </annotation>  <restriction base=**"string"**>  <minLength value=**"1"**/>  <maxLength value=**"100"**/>  </restriction>  </simpleType>  <simpleType name=**"String50SType"**>  <annotation>  <documentation>**String of up to 50 characters.**</documentation>  </annotation>  <restriction base=**"string"**>  <minLength value=**"1"**/>  <maxLength value=**"50"**/>  </restriction>  </simpleType>  <simpleType name=**"String20SType"**>  <annotation>  <documentation>**String of up to 20 characters.**</documentation>  </annotation>  <restriction base=**"string"**>  <minLength value=**"1"**/>  <maxLength value=**"20"**/>  </restriction>  </simpleType>  <simpleType name=**"Hex32SType"**>  <annotation>  <documentation>**Hexadecimal code made of 32 characters.**</documentation>  </annotation>  <restriction base=**"string"**>  <pattern value=**"[0-9a-fA-F]{32}"**/>  </restriction>  </simpleType>  <simpleType name=**"Hex512SType"**>  <annotation>  <documentation>**Hexadecimal code made of 512 characters.**</documentation>  </annotation>  <restriction base=**"string"**>  <pattern value=**"[0-9a-fA-F]{512}"**/>  </restriction>  </simpleType>  <simpleType name=**"DecimalNegSType"**>  <annotation>  <documentation>**Decimal number that can be written as negative number.**</documentation>  </annotation>  <restriction base=**"decimal"**>  <pattern value=**"-?([1-9][0-9]\*|0)\.[0-9]{2}|0"**/>  </restriction>  </simpleType>  <simpleType name=**"RegistrationCodeSType"**>  <annotation>  <documentation>**Registration code in two letters - three numbers - two letters - three number format.**</documentation>  </annotation>  <restriction base=**"string"**>  <pattern value=**"[a-z]{2}[0-9]{3}[a-z]{2}[0-9]{3}"**/>  </restriction>  </simpleType>  <simpleType name=**"DoubleNegForQuantitySType"**>  <annotation>  <documentation>**Double number for quantity fields that can be written as negative number.**</documentation>  </annotation>  <restriction base=**"double"**>  <pattern value=**"-?([1-9][0-9]\*|0)(\.[0-9]{1,3})?"**/>  </restriction>  </simpleType>  <simpleType name=**"DoubleSType"**>  <annotation>  <documentation>**Positive double.**</documentation>  </annotation>  <restriction base=**"double"**>  <minExclusive value=**"0"**></minExclusive>  </restriction>  </simpleType>  <simpleType name=**"IntSType"**>  <annotation>  <documentation>**Positive integer.**</documentation>  </annotation>  <restriction base=**"int"**>  <minExclusive value=**"0"**/>  </restriction>  </simpleType>  <simpleType name="TaxPeriodSType">  <annotation>  <documentation>Tax period composed of month and year for which it is applied, in the format MM/YYYY.</documentation>  </annotation>  <restriction base="string">  <pattern value="((0[1-9])|(1[0-2]))\/(\d{4})"/>  </restriction>  </simpleType>  <simpleType name=**"CashDepositOperationSType"**>  <annotation>  <documentation>**Cash deposit operation types.**</documentation>  </annotation>  <restriction base=**"string"**>  <enumeration value=**"INITIAL"**>  <annotation>  <documentation>**Initial deposit in the TCR.**</documentation>  </annotation>  </enumeration>  <enumeration value=**"WITHDRAW"**>  <annotation>  <documentation>**Amount of cash withdrawn from the TCR.**</documentation>  </annotation>  </enumeration>  </restriction>  </simpleType>  <simpleType name=**"IDTypeSType"**>  <annotation>  <documentation>**Identification number types.**</documentation>  </annotation>  <restriction base=**"string"**>  <enumeration value=**"TIN"**>  <annotation>  <documentation>**JMB/PIB number.**</documentation>  </annotation>  </enumeration>  <enumeration value=**"ID"**>  <annotation>  <documentation>**Personal ID number.**</documentation>  </annotation>  </enumeration>  <enumeration value=**"PASS"**>  <annotation>  <documentation>**Passport number.**</documentation>  </annotation>  </enumeration>  <enumeration value=**"VAT"**>  <annotation>  <documentation>**VAT number.**</documentation>  </annotation>  </enumeration>  <enumeration value=**"TAX"**>  <annotation>  <documentation>**TAX number.**</documentation>  </annotation>  </enumeration>  <enumeration value=**"SOC"**>  <annotation>  <documentation>**Social security number.**</documentation>  </annotation>  </enumeration>  </restriction>  </simpleType>  <simpleType name=**"FeeTypeSType"**>  <annotation>  <documentation>**Fee types.**</documentation>  </annotation>  <restriction base=**"string"**>  <enumeration value=**"PACK"**>  <annotation>  <documentation>**Packaging fee.**</documentation>  </annotation>  </enumeration>  <enumeration value=**"BOTTLE"**>  <annotation>  <documentation>**Fee for the return of glass bottles.**</documentation>  </annotation>  </enumeration>  <enumeration value=**"COMMISSION"**>  <annotation>  <documentation>**Commission for currency exchange activities.**</documentation>  </annotation>  </enumeration>  <enumeration value=**"OTHER"**>  <annotation>  <documentation>**Other fees that are not listed here.**</documentation>  </annotation>  </enumeration>  </restriction>  </simpleType>  <simpleType name=**"CorrectiveInvTypeSType"**>  <annotation>  <documentation>**Types of the corrective invoice.**</documentation>  </annotation>  <restriction base=**"string"**>  <enumeration value=**"CORRECTIVE"**>  <annotation>  <documentation>**Corrective invoice.**</documentation>  </annotation>  </enumeration>  <enumeration value=**"DEBIT"**>  <annotation>  <documentation>**Debit note.**</documentation>  </annotation>  </enumeration>  <enumeration value=**"CREDIT"**>  <annotation>  <documentation>**Credit note.**</documentation>  </annotation>  </enumeration>  </restriction>  </simpleType>  <simpleType name=**"ExemptFromVATSType"**>  <annotation>  <documentation>**Exempt from VAT types.**</documentation>  </annotation>  <restriction base=**"string"**>  <enumeration value=**"VAT\_CL17"**>  <annotation>  <documentation>**Place of service traffic.**</documentation>  </annotation>  </enumeration>  <enumeration value=**"VAT\_CL20"**>  <annotation>  <documentation>**Tax base and correction of the tax base.**</documentation>  </annotation>  </enumeration>  <enumeration value=**"VAT\_CL26"**>  <annotation>  <documentation>**Exemptions from the public interest.**</documentation>  </annotation>  </enumeration>  <enumeration value=**"VAT\_CL27"**>  <annotation>  <documentation>**Other exemptions.**</documentation>  </annotation>  </enumeration>  <enumeration value=**"VAT\_CL28"**>  <annotation>  <documentation>**Exemptions on imports of products.**</documentation>  </annotation>  </enumeration>  <enumeration value=**"VAT\_CL29"**>  <annotation>  <documentation>**Exemptions for temporary importation of products.**</documentation>  </annotation>  </enumeration>  <enumeration value=**"VAT\_CL30"**>  <annotation>  <documentation>**Special exemptions.**</documentation>  </annotation>  </enumeration>  </restriction>  </simpleType>  <simpleType name=**"ExemptFromVATSameTaxItemSType"**>  <annotation>  <documentation>**Exempt from VAT types.**</documentation>  </annotation>  <restriction base=**"string"**>  <enumeration value=**"TYPE\_1"**>  <annotation>  <documentation>**Exempt type 1.**</documentation>  </annotation>  </enumeration>  <enumeration value=**"TYPE\_2"**>  <annotation>  <documentation>**Exempt type 2.**</documentation>  </annotation>  </enumeration>  </restriction>  </simpleType>  <simpleType name=**"SubseqDelivTypeSType"**>  <annotation>  <documentation>**Type of subsequent delivery if message is delivered after issuance.**</documentation>  </annotation>  <restriction base=**"string"**>  <enumeration value=**"NOINTERNET"**>  <annotation>  <documentation>**When TCR operates in the area where there is no Internet available.**</documentation>  </annotation>  </enumeration>  <enumeration value=**"BOUNDBOOK"**>  <annotation>  <documentation>**When TCR is not working and message cannot be created with TCR.**</documentation>  </annotation>  </enumeration>  <enumeration value=**"SERVICE"**>  <annotation>  <documentation>**When there is an issue with the fiscalization service that blocks fiscalization.**</documentation>  </annotation>  </enumeration>  <enumeration value=**"TECHNICALERROR"**>  <annotation>  <documentation>**When there is a temporary technical error at TCR side that prevents successful fiscalization.**</documentation>  </annotation>  </enumeration>  </restriction>  </simpleType>  <simpleType name=**"CountryCodeSType"**>  <annotation>  <documentation>**Country codes from ISO 3166-1 alpha3 standard**</documentation>  </annotation>  <restriction base=**"string"**>  <enumeration value=**"ABW"**><annotation><documentation>**Aruba**</documentation></annotation></enumeration>  <enumeration value=**"AFG"**><annotation><documentation>**Afghanistan**</documentation></annotation></enumeration>  <enumeration value=**"AGO"**><annotation><documentation>**Angola**</documentation></annotation></enumeration>  <enumeration value=**"AIA"**><annotation><documentation>**Anguilla**</documentation></annotation></enumeration>  <enumeration value=**"ALA"**><annotation><documentation>**Aland Islands**</documentation></annotation></enumeration>  <enumeration value=**"ALB"**><annotation><documentation>**Albania**</documentation></annotation></enumeration>  <enumeration value=**"AND"**><annotation><documentation>**Andorra**</documentation></annotation></enumeration>  <enumeration value=**"ARE"**><annotation><documentation>**United Arab Emirates**</documentation></annotation></enumeration>  <enumeration value=**"ARG"**><annotation><documentation>**Argentina**</documentation></annotation></enumeration>  <enumeration value=**"ARM"**><annotation><documentation>**Armenia**</documentation></annotation></enumeration>  <enumeration value=**"ASM"**><annotation><documentation>**American Samoa**</documentation></annotation></enumeration>  <enumeration value=**"ATA"**><annotation><documentation>**Antarctica**</documentation></annotation></enumeration>  <enumeration value=**"ATF"**><annotation><documentation>**French Southern Territories**</documentation></annotation></enumeration>  <enumeration value=**"ATG"**><annotation><documentation>**Antigua and Barbuda**</documentation></annotation></enumeration>  <enumeration value=**"AUS"**><annotation><documentation>**Australia**</documentation></annotation></enumeration>  <enumeration value=**"AUT"**><annotation><documentation>**Austria**</documentation></annotation></enumeration>  <enumeration value=**"AZE"**><annotation><documentation>**Azerbaijan**</documentation></annotation></enumeration>  <enumeration value=**"BDI"**><annotation><documentation>**Burundi**</documentation></annotation></enumeration>  <enumeration value=**"BEL"**><annotation><documentation>**Belgium**</documentation></annotation></enumeration>  <enumeration value=**"BEN"**><annotation><documentation>**Benin**</documentation></annotation></enumeration>  <enumeration value=**"BES"**><annotation><documentation>**Bonaire, Sint Eustatius and Saba**</documentation></annotation></enumeration>  <enumeration value=**"BFA"**><annotation><documentation>**Burkina Faso**</documentation></annotation></enumeration>  <enumeration value=**"BGD"**><annotation><documentation>**Bangladesh**</documentation></annotation></enumeration>  <enumeration value=**"BGR"**><annotation><documentation>**Bulgaria**</documentation></annotation></enumeration>  <enumeration value=**"BHR"**><annotation><documentation>**Bahrain**</documentation></annotation></enumeration>  <enumeration value=**"BHS"**><annotation><documentation>**Bahamas**</documentation></annotation></enumeration>  <enumeration value=**"BIH"**><annotation><documentation>**Bosnia and Herzegovina**</documentation></annotation></enumeration>  <enumeration value=**"BLM"**><annotation><documentation>**Saint Barthelemy**</documentation></annotation></enumeration>  <enumeration value=**"BLR"**><annotation><documentation>**Belarus**</documentation></annotation></enumeration>  <enumeration value=**"BLZ"**><annotation><documentation>**Belize**</documentation></annotation></enumeration>  <enumeration value=**"BMU"**><annotation><documentation>**Bermuda**</documentation></annotation></enumeration>  <enumeration value=**"BOL"**><annotation><documentation>**Bolivia (Plurinational State of)**</documentation></annotation></enumeration>  <enumeration value=**"BRA"**><annotation><documentation>**Brazil**</documentation></annotation></enumeration>  <enumeration value=**"BRB"**><annotation><documentation>**Barbados**</documentation></annotation></enumeration>  <enumeration value=**"BRN"**><annotation><documentation>**Brunei Darussalam**</documentation></annotation></enumeration>  <enumeration value=**"BTN"**><annotation><documentation>**Bhutan**</documentation></annotation></enumeration>  <enumeration value=**"BVT"**><annotation><documentation>**Bouvet Island**</documentation></annotation></enumeration>  <enumeration value=**"BWA"**><annotation><documentation>**Botswana**</documentation></annotation></enumeration>  <enumeration value=**"CAF"**><annotation><documentation>**Central African Republic**</documentation></annotation></enumeration>  <enumeration value=**"CAN"**><annotation><documentation>**Canada**</documentation></annotation></enumeration>  <enumeration value=**"CCK"**><annotation><documentation>**Cocos (Keeling) Islands**</documentation></annotation></enumeration>  <enumeration value=**"CHE"**><annotation><documentation>**Switzerland**</documentation></annotation></enumeration>  <enumeration value=**"CHL"**><annotation><documentation>**Chile**</documentation></annotation></enumeration>  <enumeration value=**"CHN"**><annotation><documentation>**China**</documentation></annotation></enumeration>  <enumeration value=**"CIV"**><annotation><documentation>**Cote d'Ivoire**</documentation></annotation></enumeration>  <enumeration value=**"CMR"**><annotation><documentation>**Cameroon**</documentation></annotation></enumeration>  <enumeration value=**"COD"**><annotation><documentation>**Congo, Democratic Republic of the**</documentation></annotation></enumeration>  <enumeration value=**"COG"**><annotation><documentation>**Congo**</documentation></annotation></enumeration>  <enumeration value=**"COK"**><annotation><documentation>**Cook Islands**</documentation></annotation></enumeration>  <enumeration value=**"COL"**><annotation><documentation>**Colombia**</documentation></annotation></enumeration>  <enumeration value=**"COM"**><annotation><documentation>**Comoros**</documentation></annotation></enumeration>  <enumeration value=**"CPV"**><annotation><documentation>**Cabo Verde**</documentation></annotation></enumeration>  <enumeration value=**"CRI"**><annotation><documentation>**Costa Rica**</documentation></annotation></enumeration>  <enumeration value=**"CUB"**><annotation><documentation>**Cuba**</documentation></annotation></enumeration>  <enumeration value=**"CUW"**><annotation><documentation>**Curacao**</documentation></annotation></enumeration>  <enumeration value=**"CXR"**><annotation><documentation>**Christmas Island**</documentation></annotation></enumeration>  <enumeration value=**"CYM"**><annotation><documentation>**Cayman Islands**</documentation></annotation></enumeration>  <enumeration value=**"CYP"**><annotation><documentation>**Cyprus**</documentation></annotation></enumeration>  <enumeration value=**"CZE"**><annotation><documentation>**Czechia**</documentation></annotation></enumeration>  <enumeration value=**"DEU"**><annotation><documentation>**Germany**</documentation></annotation></enumeration>  <enumeration value=**"DJI"**><annotation><documentation>**Djibouti**</documentation></annotation></enumeration>  <enumeration value=**"DMA"**><annotation><documentation>**Dominica**</documentation></annotation></enumeration>  <enumeration value=**"DNK"**><annotation><documentation>**Denmark**</documentation></annotation></enumeration>  <enumeration value=**"DOM"**><annotation><documentation>**Dominican Republic**</documentation></annotation></enumeration>  <enumeration value=**"DZA"**><annotation><documentation>**Algeria**</documentation></annotation></enumeration>  <enumeration value=**"ECU"**><annotation><documentation>**Ecuador**</documentation></annotation></enumeration>  <enumeration value=**"EGY"**><annotation><documentation>**Egypt**</documentation></annotation></enumeration>  <enumeration value=**"ERI"**><annotation><documentation>**Eritrea**</documentation></annotation></enumeration>  <enumeration value=**"ESH"**><annotation><documentation>**Western Sahara**</documentation></annotation></enumeration>  <enumeration value=**"ESP"**><annotation><documentation>**Spain**</documentation></annotation></enumeration>  <enumeration value=**"EST"**><annotation><documentation>**Estonia**</documentation></annotation></enumeration>  <enumeration value=**"ETH"**><annotation><documentation>**Ethiopia**</documentation></annotation></enumeration>  <enumeration value=**"FIN"**><annotation><documentation>**Finland**</documentation></annotation></enumeration>  <enumeration value=**"FJI"**><annotation><documentation>**Fiji**</documentation></annotation></enumeration>  <enumeration value=**"FLK"**><annotation><documentation>**Falkland Islands (Malvinas)**</documentation></annotation></enumeration>  <enumeration value=**"FRA"**><annotation><documentation>**France**</documentation></annotation></enumeration>  <enumeration value=**"FRO"**><annotation><documentation>**Faroe Islands**</documentation></annotation></enumeration>  <enumeration value=**"FSM"**><annotation><documentation>**Micronesia (Federated States of)**</documentation></annotation></enumeration>  <enumeration value=**"GAB"**><annotation><documentation>**Gabon**</documentation></annotation></enumeration>  <enumeration value=**"GBR"**><annotation><documentation>**United Kingdom of Great Britain and Northern Ireland**</documentation></annotation></enumeration>  <enumeration value=**"GEO"**><annotation><documentation>**Georgia**</documentation></annotation></enumeration>  <enumeration value=**"GGY"**><annotation><documentation>**Guernsey**</documentation></annotation></enumeration>  <enumeration value=**"GHA"**><annotation><documentation>**Ghana**</documentation></annotation></enumeration>  <enumeration value=**"GIB"**><annotation><documentation>**Gibraltar**</documentation></annotation></enumeration>  <enumeration value=**"GIN"**><annotation><documentation>**Guinea**</documentation></annotation></enumeration>  <enumeration value=**"GLP"**><annotation><documentation>**Guadeloupe**</documentation></annotation></enumeration>  <enumeration value=**"GMB"**><annotation><documentation>**Gambia**</documentation></annotation></enumeration>  <enumeration value=**"GNB"**><annotation><documentation>**Guinea-Bissau**</documentation></annotation></enumeration>  <enumeration value=**"GNQ"**><annotation><documentation>**Equatorial Guinea**</documentation></annotation></enumeration>  <enumeration value=**"GRC"**><annotation><documentation>**Greece**</documentation></annotation></enumeration>  <enumeration value=**"GRD"**><annotation><documentation>**Grenada**</documentation></annotation></enumeration>  <enumeration value=**"GRL"**><annotation><documentation>**Greenland**</documentation></annotation></enumeration>  <enumeration value=**"GTM"**><annotation><documentation>**Guatemala**</documentation></annotation></enumeration>  <enumeration value=**"GUF"**><annotation><documentation>**French Guiana**</documentation></annotation></enumeration>  <enumeration value=**"GUM"**><annotation><documentation>**Guam**</documentation></annotation></enumeration>  <enumeration value=**"GUY"**><annotation><documentation>**Guyana**</documentation></annotation></enumeration>  <enumeration value=**"HKG"**><annotation><documentation>**Hong Kong**</documentation></annotation></enumeration>  <enumeration value=**"HMD"**><annotation><documentation>**Heard Island and McDonald Islands**</documentation></annotation></enumeration>  <enumeration value=**"HND"**><annotation><documentation>**Honduras**</documentation></annotation></enumeration>  <enumeration value=**"HRV"**><annotation><documentation>**Croatia**</documentation></annotation></enumeration>  <enumeration value=**"HTI"**><annotation><documentation>**Haiti**</documentation></annotation></enumeration>  <enumeration value=**"HUN"**><annotation><documentation>**Hungary**</documentation></annotation></enumeration>  <enumeration value=**"IDN"**><annotation><documentation>**Indonesia**</documentation></annotation></enumeration>  <enumeration value=**"IMN"**><annotation><documentation>**Isle of Man**</documentation></annotation></enumeration>  <enumeration value=**"IND"**><annotation><documentation>**India**</documentation></annotation></enumeration>  <enumeration value=**"IOT"**><annotation><documentation>**British Indian Ocean Territory**</documentation></annotation></enumeration>  <enumeration value=**"IRL"**><annotation><documentation>**Ireland**</documentation></annotation></enumeration>  <enumeration value=**"IRN"**><annotation><documentation>**Iran (Islamic Republic of)**</documentation></annotation></enumeration>  <enumeration value=**"IRQ"**><annotation><documentation>**Iraq**</documentation></annotation></enumeration>  <enumeration value=**"ISL"**><annotation><documentation>**Iceland**</documentation></annotation></enumeration>  <enumeration value=**"ISR"**><annotation><documentation>**Israel**</documentation></annotation></enumeration>  <enumeration value=**"ITA"**><annotation><documentation>**Italy**</documentation></annotation></enumeration>  <enumeration value=**"JAM"**><annotation><documentation>**Jamaica**</documentation></annotation></enumeration>  <enumeration value=**"JEY"**><annotation><documentation>**Jersey**</documentation></annotation></enumeration>  <enumeration value=**"JOR"**><annotation><documentation>**Jordan**</documentation></annotation></enumeration>  <enumeration value=**"JPN"**><annotation><documentation>**Japan**</documentation></annotation></enumeration>  <enumeration value=**"KAZ"**><annotation><documentation>**Kazakhstan**</documentation></annotation></enumeration>  <enumeration value=**"KEN"**><annotation><documentation>**Kenya**</documentation></annotation></enumeration>  <enumeration value=**"KGZ"**><annotation><documentation>**Kyrgyzstan**</documentation></annotation></enumeration>  <enumeration value=**"KHM"**><annotation><documentation>**Cambodia**</documentation></annotation></enumeration>  <enumeration value=**"KIR"**><annotation><documentation>**Kiribati**</documentation></annotation></enumeration>  <enumeration value=**"KNA"**><annotation><documentation>**Saint Kitts and Nevis**</documentation></annotation></enumeration>  <enumeration value=**"KOR"**><annotation><documentation>**Korea, Republic of**</documentation></annotation></enumeration>  <enumeration value=**"KWT"**><annotation><documentation>**Kuwait**</documentation></annotation></enumeration>  <enumeration value=**"LAO"**><annotation><documentation>**Lao People's Democratic Republic**</documentation></annotation></enumeration>  <enumeration value=**"LBN"**><annotation><documentation>**Lebanon**</documentation></annotation></enumeration>  <enumeration value=**"LBR"**><annotation><documentation>**Liberia**</documentation></annotation></enumeration>  <enumeration value=**"LBY"**><annotation><documentation>**Libya**</documentation></annotation></enumeration>  <enumeration value=**"LCA"**><annotation><documentation>**Saint Lucia**</documentation></annotation></enumeration>  <enumeration value=**"LIE"**><annotation><documentation>**Liechtenstein**</documentation></annotation></enumeration>  <enumeration value=**"LKA"**><annotation><documentation>**Sri Lanka**</documentation></annotation></enumeration>  <enumeration value=**"LSO"**><annotation><documentation>**Lesotho**</documentation></annotation></enumeration>  <enumeration value=**"LTU"**><annotation><documentation>**Lithuania**</documentation></annotation></enumeration>  <enumeration value=**"LUX"**><annotation><documentation>**Luxembourg**</documentation></annotation></enumeration>  <enumeration value=**"LVA"**><annotation><documentation>**Latvia**</documentation></annotation></enumeration>  <enumeration value=**"MAC"**><annotation><documentation>**Macao**</documentation></annotation></enumeration>  <enumeration value=**"MAF"**><annotation><documentation>**Saint Martin (French part)**</documentation></annotation></enumeration>  <enumeration value=**"MAR"**><annotation><documentation>**Morocco**</documentation></annotation></enumeration>  <enumeration value=**"MCO"**><annotation><documentation>**Monaco**</documentation></annotation></enumeration>  <enumeration value=**"MDA"**><annotation><documentation>**Moldova, Republic of**</documentation></annotation></enumeration>  <enumeration value=**"MDG"**><annotation><documentation>**Madagascar**</documentation></annotation></enumeration>  <enumeration value=**"MDV"**><annotation><documentation>**Maldives**</documentation></annotation></enumeration>  <enumeration value=**"MEX"**><annotation><documentation>**Mexico**</documentation></annotation></enumeration>  <enumeration value=**"MHL"**><annotation><documentation>**Marshall Islands**</documentation></annotation></enumeration>  <enumeration value=**"MKD"**><annotation><documentation>**North Macedonia**</documentation></annotation></enumeration>  <enumeration value=**"MLI"**><annotation><documentation>**Mali**</documentation></annotation></enumeration>  <enumeration value=**"MLT"**><annotation><documentation>**Malta**</documentation></annotation></enumeration>  <enumeration value=**"MMR"**><annotation><documentation>**Myanmar**</documentation></annotation></enumeration>  <enumeration value=**"MNE"**><annotation><documentation>**Montenegro**</documentation></annotation></enumeration>  <enumeration value=**"MNG"**><annotation><documentation>**Mongolia**</documentation></annotation></enumeration>  <enumeration value=**"MNP"**><annotation><documentation>**Northern Mariana Islands**</documentation></annotation></enumeration>  <enumeration value=**"MOZ"**><annotation><documentation>**Mozambique**</documentation></annotation></enumeration>  <enumeration value=**"MRT"**><annotation><documentation>**Mauritania**</documentation></annotation></enumeration>  <enumeration value=**"MSR"**><annotation><documentation>**Montserrat**</documentation></annotation></enumeration>  <enumeration value=**"MTQ"**><annotation><documentation>**Martinique**</documentation></annotation></enumeration>  <enumeration value=**"MUS"**><annotation><documentation>**Mauritius**</documentation></annotation></enumeration>  <enumeration value=**"MWI"**><annotation><documentation>**Malawi**</documentation></annotation></enumeration>  <enumeration value=**"MYS"**><annotation><documentation>**Malaysia**</documentation></annotation></enumeration>  <enumeration value=**"MYT"**><annotation><documentation>**Mayotte**</documentation></annotation></enumeration>  <enumeration value=**"NAM"**><annotation><documentation>**Namibia**</documentation></annotation></enumeration>  <enumeration value=**"NCL"**><annotation><documentation>**New Caledonia**</documentation></annotation></enumeration>  <enumeration value=**"NER"**><annotation><documentation>**Niger**</documentation></annotation></enumeration>  <enumeration value=**"NFK"**><annotation><documentation>**Norfolk Island**</documentation></annotation></enumeration>  <enumeration value=**"NGA"**><annotation><documentation>**Nigeria**</documentation></annotation></enumeration>  <enumeration value=**"NIC"**><annotation><documentation>**Nicaragua**</documentation></annotation></enumeration>  <enumeration value=**"NIU"**><annotation><documentation>**Niue**</documentation></annotation></enumeration>  <enumeration value=**"NLD"**><annotation><documentation>**Netherlands**</documentation></annotation></enumeration>  <enumeration value=**"NOR"**><annotation><documentation>**Norway**</documentation></annotation></enumeration>  <enumeration value=**"NPL"**><annotation><documentation>**Nepal**</documentation></annotation></enumeration>  <enumeration value=**"NRU"**><annotation><documentation>**Nauru**</documentation></annotation></enumeration>  <enumeration value=**"NZL"**><annotation><documentation>**New Zealand**</documentation></annotation></enumeration>  <enumeration value=**"OMN"**><annotation><documentation>**Oman**</documentation></annotation></enumeration>  <enumeration value=**"PAK"**><annotation><documentation>**Pakistan**</documentation></annotation></enumeration>  <enumeration value=**"PAN"**><annotation><documentation>**Panama**</documentation></annotation></enumeration>  <enumeration value=**"PCN"**><annotation><documentation>**Pitcairn**</documentation></annotation></enumeration>  <enumeration value=**"PER"**><annotation><documentation>**Peru**</documentation></annotation></enumeration>  <enumeration value=**"PHL"**><annotation><documentation>**Philippines**</documentation></annotation></enumeration>  <enumeration value=**"PLW"**><annotation><documentation>**Palau**</documentation></annotation></enumeration>  <enumeration value=**"PNG"**><annotation><documentation>**Papua New Guinea**</documentation></annotation></enumeration>  <enumeration value=**"POL"**><annotation><documentation>**Poland**</documentation></annotation></enumeration>  <enumeration value=**"PRI"**><annotation><documentation>**Puerto Rico**</documentation></annotation></enumeration>  <enumeration value=**"PRK"**><annotation><documentation>**Korea (Democratic People's Republic of)**</documentation></annotation></enumeration>  <enumeration value=**"PRT"**><annotation><documentation>**Portugal**</documentation></annotation></enumeration>  <enumeration value=**"PRY"**><annotation><documentation>**Paraguay**</documentation></annotation></enumeration>  <enumeration value=**"PSE"**><annotation><documentation>**Palestine, State of**</documentation></annotation></enumeration>  <enumeration value=**"PYF"**><annotation><documentation>**French Polynesia**</documentation></annotation></enumeration>  <enumeration value=**"QAT"**><annotation><documentation>**Qatar**</documentation></annotation></enumeration>  <enumeration value=**"REU"**><annotation><documentation>**Reunion**</documentation></annotation></enumeration>  <enumeration value=**"ROU"**><annotation><documentation>**Romania**</documentation></annotation></enumeration>  <enumeration value=**"RUS"**><annotation><documentation>**Russian Federation**</documentation></annotation></enumeration>  <enumeration value=**"RWA"**><annotation><documentation>**Rwanda**</documentation></annotation></enumeration>  <enumeration value=**"SAU"**><annotation><documentation>**Saudi Arabia**</documentation></annotation></enumeration>  <enumeration value=**"SDN"**><annotation><documentation>**Sudan**</documentation></annotation></enumeration>  <enumeration value=**"SEN"**><annotation><documentation>**Senegal**</documentation></annotation></enumeration>  <enumeration value=**"SGP"**><annotation><documentation>**Singapore**</documentation></annotation></enumeration>  <enumeration value=**"SGS"**><annotation><documentation>**South Georgia and the South Sandwich Islands**</documentation></annotation></enumeration>  <enumeration value=**"SHN"**><annotation><documentation>**Saint Helena, Ascension and Tristan da Cunha**</documentation></annotation></enumeration>  <enumeration value=**"SJM"**><annotation><documentation>**Svalbard and Jan Mayen**</documentation></annotation></enumeration>  <enumeration value=**"SLB"**><annotation><documentation>**Solomon Islands**</documentation></annotation></enumeration>  <enumeration value=**"SLE"**><annotation><documentation>**Sierra Leone**</documentation></annotation></enumeration>  <enumeration value=**"SLV"**><annotation><documentation>**El Salvador**</documentation></annotation></enumeration>  <enumeration value=**"SMR"**><annotation><documentation>**San Marino**</documentation></annotation></enumeration>  <enumeration value=**"SOM"**><annotation><documentation>**Somalia**</documentation></annotation></enumeration>  <enumeration value=**"SPM"**><annotation><documentation>**Saint Pierre and Miquelon**</documentation></annotation></enumeration>  <enumeration value=**"SRB"**><annotation><documentation>**Serbia**</documentation></annotation></enumeration>  <enumeration value=**"SSD"**><annotation><documentation>**South Sudan**</documentation></annotation></enumeration>  <enumeration value=**"STP"**><annotation><documentation>**Sao Tome and Principe**</documentation></annotation></enumeration>  <enumeration value=**"SUR"**><annotation><documentation>**Suriname**</documentation></annotation></enumeration>  <enumeration value=**"SVK"**><annotation><documentation>**Slovakia**</documentation></annotation></enumeration>  <enumeration value=**"SVN"**><annotation><documentation>**Slovenia**</documentation></annotation></enumeration>  <enumeration value=**"SWE"**><annotation><documentation>**Sweden**</documentation></annotation></enumeration>  <enumeration value=**"SWZ"**><annotation><documentation>**Eswatini**</documentation></annotation></enumeration>  <enumeration value=**"SXM"**><annotation><documentation>**Sint Maarten (Dutch part)**</documentation></annotation></enumeration>  <enumeration value=**"SYC"**><annotation><documentation>**Seychelles**</documentation></annotation></enumeration>  <enumeration value=**"SYR"**><annotation><documentation>**Syrian Arab Republic**</documentation></annotation></enumeration>  <enumeration value=**"TCA"**><annotation><documentation>**Turks and Caicos Islands**</documentation></annotation></enumeration>  <enumeration value=**"TCD"**><annotation><documentation>**Chad**</documentation></annotation></enumeration>  <enumeration value=**"TGO"**><annotation><documentation>**Togo**</documentation></annotation></enumeration>  <enumeration value=**"THA"**><annotation><documentation>**Thailand**</documentation></annotation></enumeration>  <enumeration value=**"TJK"**><annotation><documentation>**Tajikistan**</documentation></annotation></enumeration>  <enumeration value=**"TKL"**><annotation><documentation>**Tokelau**</documentation></annotation></enumeration>  <enumeration value=**"TKM"**><annotation><documentation>**Turkmenistan**</documentation></annotation></enumeration>  <enumeration value=**"TLS"**><annotation><documentation>**Timor-Leste**</documentation></annotation></enumeration>  <enumeration value=**"TON"**><annotation><documentation>**Tonga**</documentation></annotation></enumeration>  <enumeration value=**"TTO"**><annotation><documentation>**Trinidad and Tobago**</documentation></annotation></enumeration>  <enumeration value=**"TUN"**><annotation><documentation>**Tunisia**</documentation></annotation></enumeration>  <enumeration value=**"TUR"**><annotation><documentation>**Turkey**</documentation></annotation></enumeration>  <enumeration value=**"TUV"**><annotation><documentation>**Tuvalu**</documentation></annotation></enumeration>  <enumeration value=**"TWN"**><annotation><documentation>**Taiwan, Province of China**</documentation></annotation></enumeration>  <enumeration value=**"TZA"**><annotation><documentation>**Tanzania, United Republic of**</documentation></annotation></enumeration>  <enumeration value=**"UGA"**><annotation><documentation>**Uganda**</documentation></annotation></enumeration>  <enumeration value=**"UKR"**><annotation><documentation>**Ukraine**</documentation></annotation></enumeration>  <enumeration value=**"UMI"**><annotation><documentation>**United States Minor Outlying Islands**</documentation></annotation></enumeration>  <enumeration value=**"RKS"**><annotation><documentation>**Kosovo**</documentation></annotation></enumeration>  <enumeration value=**"URY"**><annotation><documentation>**Uruguay**</documentation></annotation></enumeration>  <enumeration value=**"USA"**><annotation><documentation>**United States of America**</documentation></annotation></enumeration>  <enumeration value=**"UZB"**><annotation><documentation>**Uzbekistan**</documentation></annotation></enumeration>  <enumeration value=**"VAT"**><annotation><documentation>**Holy See**</documentation></annotation></enumeration>  <enumeration value=**"VCT"**><annotation><documentation>**Saint Vincent and the Grenadines**</documentation></annotation></enumeration>  <enumeration value=**"VEN"**><annotation><documentation>**Venezuela (Bolivarian Republic of)**</documentation></annotation></enumeration>  <enumeration value=**"VGB"**><annotation><documentation>**Virgin Islands (British)**</documentation></annotation></enumeration>  <enumeration value=**"VIR"**><annotation><documentation>**Virgin Islands (U.S.)**</documentation></annotation></enumeration>  <enumeration value=**"VNM"**><annotation><documentation>**Viet Nam**</documentation></annotation></enumeration>  <enumeration value=**"VUT"**><annotation><documentation>**Vanuatu**</documentation></annotation></enumeration>  <enumeration value=**"WLF"**><annotation><documentation>**Wallis and Futuna**</documentation></annotation></enumeration>  <enumeration value=**"WSM"**><annotation><documentation>**Samoa**</documentation></annotation></enumeration>  <enumeration value=**"YEM"**><annotation><documentation>**Yemen**</documentation></annotation></enumeration>  <enumeration value=**"ZAF"**><annotation><documentation>**South Africa**</documentation></annotation></enumeration>  <enumeration value=**"ZMB"**><annotation><documentation>**Zambia**</documentation></annotation></enumeration>  <enumeration value=**"ZWE"**><annotation><documentation>**Zimbabwe**</documentation></annotation></enumeration>  </restriction>  </simpleType>  <simpleType name=**"CurrencyCodeSType"**>  <annotation>  <documentation>**Currency codes from ISO 4217 standard**</documentation>  </annotation>  <restriction base=**"string"**>  <enumeration value=**"AED"**><annotation><documentation>**United Arab Emirates Dirham**</documentation></annotation></enumeration>  <enumeration value=**"AFN"**><annotation><documentation>**Afghanistan Afghani**</documentation></annotation></enumeration>  <enumeration value=**"AMD"**><annotation><documentation>**Armenia Dram**</documentation></annotation></enumeration>  <enumeration value=**"ANG"**><annotation><documentation>**Netherlands Antilles Guilder**</documentation></annotation></enumeration>  <enumeration value=**"AOA"**><annotation><documentation>**Angola Kwanza**</documentation></annotation></enumeration>  <enumeration value=**"ARS"**><annotation><documentation>**Argentina Peso**</documentation></annotation></enumeration>  <enumeration value=**"AUD"**><annotation><documentation>**Australia Dollar**</documentation></annotation></enumeration>  <enumeration value=**"AWG"**><annotation><documentation>**Aruba Guilder**</documentation></annotation></enumeration>  <enumeration value=**"AZN"**><annotation><documentation>**Azerbaijan Manat**</documentation></annotation></enumeration>  <enumeration value=**"BAM"**><annotation><documentation>**Bosnia and Herzegovina Convertible Mark**</documentation></annotation></enumeration>  <enumeration value=**"BBD"**><annotation><documentation>**Barbados Dollar**</documentation></annotation></enumeration>  <enumeration value=**"BDT"**><annotation><documentation>**Bangladesh Taka**</documentation></annotation></enumeration>  <enumeration value=**"BGN"**><annotation><documentation>**Bulgaria Lev**</documentation></annotation></enumeration>  <enumeration value=**"BHD"**><annotation><documentation>**Bahrain Dinar**</documentation></annotation></enumeration>  <enumeration value=**"BIF"**><annotation><documentation>**Burundi Franc**</documentation></annotation></enumeration>  <enumeration value=**"BMD"**><annotation><documentation>**Bermuda Dollar**</documentation></annotation></enumeration>  <enumeration value=**"BND"**><annotation><documentation>**Brunei Darussalam Dollar**</documentation></annotation></enumeration>  <enumeration value=**"BOB"**><annotation><documentation>**Bolivia Boliviano**</documentation></annotation></enumeration>  <enumeration value=**"BRL"**><annotation><documentation>**Brazil Real**</documentation></annotation></enumeration>  <enumeration value=**"BSD"**><annotation><documentation>**Bahamas Dollar**</documentation></annotation></enumeration>  <enumeration value=**"BTN"**><annotation><documentation>**Bhutan Ngultrum**</documentation></annotation></enumeration>  <enumeration value=**"BWP"**><annotation><documentation>**Botswana Pula**</documentation></annotation></enumeration>  <enumeration value=**"BYN"**><annotation><documentation>**Belarus Ruble**</documentation></annotation></enumeration>  <enumeration value=**"BZD"**><annotation><documentation>**Belize Dollar**</documentation></annotation></enumeration>  <enumeration value=**"CAD"**><annotation><documentation>**Canada Dollar**</documentation></annotation></enumeration>  <enumeration value=**"CDF"**><annotation><documentation>**Congo/Kinshasa Franc**</documentation></annotation></enumeration>  <enumeration value=**"CHF"**><annotation><documentation>**Switzerland Franc**</documentation></annotation></enumeration>  <enumeration value=**"CLP"**><annotation><documentation>**Chile Peso**</documentation></annotation></enumeration>  <enumeration value=**"CNY"**><annotation><documentation>**China Yuan Renminbi**</documentation></annotation></enumeration>  <enumeration value=**"COP"**><annotation><documentation>**Colombia Peso**</documentation></annotation></enumeration>  <enumeration value=**"CRC"**><annotation><documentation>**Costa Rica Colon**</documentation></annotation></enumeration>  <enumeration value=**"CUC"**><annotation><documentation>**Cuba Convertible Peso**</documentation></annotation></enumeration>  <enumeration value=**"CUP"**><annotation><documentation>**Cuba Peso**</documentation></annotation></enumeration>  <enumeration value=**"CVE"**><annotation><documentation>**Cape Verde Escudo**</documentation></annotation></enumeration>  <enumeration value=**"CZK"**><annotation><documentation>**Czech Republic Koruna**</documentation></annotation></enumeration>  <enumeration value=**"DJF"**><annotation><documentation>**Djibouti Franc**</documentation></annotation></enumeration>  <enumeration value=**"DKK"**><annotation><documentation>**Denmark Krone**</documentation></annotation></enumeration>  <enumeration value=**"DOP"**><annotation><documentation>**Dominican Republic Peso**</documentation></annotation></enumeration>  <enumeration value=**"DZD"**><annotation><documentation>**Algeria Dinar**</documentation></annotation></enumeration>  <enumeration value=**"EGP"**><annotation><documentation>**Egypt Pound**</documentation></annotation></enumeration>  <enumeration value=**"ERN"**><annotation><documentation>**Eritrea Nakfa**</documentation></annotation></enumeration>  <enumeration value=**"ETB"**><annotation><documentation>**Ethiopia Birr**</documentation></annotation></enumeration>  <enumeration value=**"EUR"**><annotation><documentation>**Euro Member Countries**</documentation></annotation></enumeration>  <enumeration value=**"FJD"**><annotation><documentation>**Fiji Dollar**</documentation></annotation></enumeration>  <enumeration value=**"FKP"**><annotation><documentation>**Falkland Islands (Malvinas) Pound**</documentation></annotation></enumeration>  <enumeration value=**"GBP"**><annotation><documentation>**United Kingdom Pound**</documentation></annotation></enumeration>  <enumeration value=**"GEL"**><annotation><documentation>**Georgia Lari**</documentation></annotation></enumeration>  <enumeration value=**"GGP"**><annotation><documentation>**Guernsey Pound**</documentation></annotation></enumeration>  <enumeration value=**"GHS"**><annotation><documentation>**Ghana Cedi**</documentation></annotation></enumeration>  <enumeration value=**"GIP"**><annotation><documentation>**Gibraltar Pound**</documentation></annotation></enumeration>  <enumeration value=**"GMD"**><annotation><documentation>**Gambia Dalasi**</documentation></annotation></enumeration>  <enumeration value=**"GNF"**><annotation><documentation>**Guinea Franc**</documentation></annotation></enumeration>  <enumeration value=**"GTQ"**><annotation><documentation>**Guatemala Quetzal**</documentation></annotation></enumeration>  <enumeration value=**"GYD"**><annotation><documentation>**Guyana Dollar**</documentation></annotation></enumeration>  <enumeration value=**"HKD"**><annotation><documentation>**Hong Kong Dollar**</documentation></annotation></enumeration>  <enumeration value=**"HNL"**><annotation><documentation>**Honduras Lempira**</documentation></annotation></enumeration>  <enumeration value=**"HRK"**><annotation><documentation>**Croatia Kuna**</documentation></annotation></enumeration>  <enumeration value=**"HTG"**><annotation><documentation>**Haiti Gourde**</documentation></annotation></enumeration>  <enumeration value=**"HUF"**><annotation><documentation>**Hungary Forint**</documentation></annotation></enumeration>  <enumeration value=**"IDR"**><annotation><documentation>**Indonesia Rupiah**</documentation></annotation></enumeration>  <enumeration value=**"ILS"**><annotation><documentation>**Israel Shekel**</documentation></annotation></enumeration>  <enumeration value=**"IMP"**><annotation><documentation>**Isle of Man Pound**</documentation></annotation></enumeration>  <enumeration value=**"INR"**><annotation><documentation>**India Rupee**</documentation></annotation></enumeration>  <enumeration value=**"IQD"**><annotation><documentation>**Iraq Dinar**</documentation></annotation></enumeration>  <enumeration value=**"IRR"**><annotation><documentation>**Iran Rial**</documentation></annotation></enumeration>  <enumeration value=**"ISK"**><annotation><documentation>**Iceland Krona**</documentation></annotation></enumeration>  <enumeration value=**"JEP"**><annotation><documentation>**Jersey Pound**</documentation></annotation></enumeration>  <enumeration value=**"JMD"**><annotation><documentation>**Jamaica Dollar**</documentation></annotation></enumeration>  <enumeration value=**"JOD"**><annotation><documentation>**Jordan Dinar**</documentation></annotation></enumeration>  <enumeration value=**"JPY"**><annotation><documentation>**Japan Yen**</documentation></annotation></enumeration>  <enumeration value=**"KES"**><annotation><documentation>**Kenya Shilling**</documentation></annotation></enumeration>  <enumeration value=**"KGS"**><annotation><documentation>**Kyrgyzstan Som**</documentation></annotation></enumeration>  <enumeration value=**"KHR"**><annotation><documentation>**Cambodia Riel**</documentation></annotation></enumeration>  <enumeration value=**"KMF"**><annotation><documentation>**Comorian Franc**</documentation></annotation></enumeration>  <enumeration value=**"KPW"**><annotation><documentation>**Korea (North) Won**</documentation></annotation></enumeration>  <enumeration value=**"KRW"**><annotation><documentation>**Korea (South) Won**</documentation></annotation></enumeration>  <enumeration value=**"KWD"**><annotation><documentation>**Kuwait Dinar**</documentation></annotation></enumeration>  <enumeration value=**"KYD"**><annotation><documentation>**Cayman Islands Dollar**</documentation></annotation></enumeration>  <enumeration value=**"KZT"**><annotation><documentation>**Kazakhstan Tenge**</documentation></annotation></enumeration>  <enumeration value=**"LAK"**><annotation><documentation>**Laos Kip**</documentation></annotation></enumeration>  <enumeration value=**"LBP"**><annotation><documentation>**Lebanon Pound**</documentation></annotation></enumeration>  <enumeration value=**"LKR"**><annotation><documentation>**Sri Lanka Rupee**</documentation></annotation></enumeration>  <enumeration value=**"LRD"**><annotation><documentation>**Liberia Dollar**</documentation></annotation></enumeration>  <enumeration value=**"LSL"**><annotation><documentation>**Lesotho Loti**</documentation></annotation></enumeration>  <enumeration value=**"LYD"**><annotation><documentation>**Libya Dinar**</documentation></annotation></enumeration>  <enumeration value=**"MAD"**><annotation><documentation>**Morocco Dirham**</documentation></annotation></enumeration>  <enumeration value=**"MDL"**><annotation><documentation>**Moldova Leu**</documentation></annotation></enumeration>  <enumeration value=**"MGA"**><annotation><documentation>**Madagascar Ariary**</documentation></annotation></enumeration>  <enumeration value=**"MKD"**><annotation><documentation>**Macedonia Denar**</documentation></annotation></enumeration>  <enumeration value=**"MMK"**><annotation><documentation>**Myanmar (Burma) Kyat**</documentation></annotation></enumeration>  <enumeration value=**"MNT"**><annotation><documentation>**Mongolia Tughrik**</documentation></annotation></enumeration>  <enumeration value=**"MOP"**><annotation><documentation>**Macau Pataca**</documentation></annotation></enumeration>  <enumeration value=**"MRU"**><annotation><documentation>**Mauritania Ouguiya**</documentation></annotation></enumeration>  <enumeration value=**"MUR"**><annotation><documentation>**Mauritius Rupee**</documentation></annotation></enumeration>  <enumeration value=**"MVR"**><annotation><documentation>**Maldives (Maldive Islands) Rufiyaa**</documentation></annotation></enumeration>  <enumeration value=**"MWK"**><annotation><documentation>**Malawi Kwacha**</documentation></annotation></enumeration>  <enumeration value=**"MXN"**><annotation><documentation>**Mexico Peso**</documentation></annotation></enumeration>  <enumeration value=**"MYR"**><annotation><documentation>**Malaysia Ringgit**</documentation></annotation></enumeration>  <enumeration value=**"MZN"**><annotation><documentation>**Mozambique Metical**</documentation></annotation></enumeration>  <enumeration value=**"NAD"**><annotation><documentation>**Namibia Dollar**</documentation></annotation></enumeration>  <enumeration value=**"NGN"**><annotation><documentation>**Nigeria Naira**</documentation></annotation></enumeration>  <enumeration value=**"NIO"**><annotation><documentation>**Nicaragua Cordoba**</documentation></annotation></enumeration>  <enumeration value=**"NOK"**><annotation><documentation>**Norway Krone**</documentation></annotation></enumeration>  <enumeration value=**"NPR"**><annotation><documentation>**Nepal Rupee**</documentation></annotation></enumeration>  <enumeration value=**"NZD"**><annotation><documentation>**New Zealand Dollar**</documentation></annotation></enumeration>  <enumeration value=**"OMR"**><annotation><documentation>**Oman Rial**</documentation></annotation></enumeration>  <enumeration value=**"PAB"**><annotation><documentation>**Panama Balboa**</documentation></annotation></enumeration>  <enumeration value=**"PEN"**><annotation><documentation>**Peru Sol**</documentation></annotation></enumeration>  <enumeration value=**"PGK"**><annotation><documentation>**Papua New Guinea Kina**</documentation></annotation></enumeration>  <enumeration value=**"PHP"**><annotation><documentation>**Philippines Peso**</documentation></annotation></enumeration>  <enumeration value=**"PKR"**><annotation><documentation>**Pakistan Rupee**</documentation></annotation></enumeration>  <enumeration value=**"PLN"**><annotation><documentation>**Poland Zloty**</documentation></annotation></enumeration>  <enumeration value=**"PYG"**><annotation><documentation>**Paraguay Guarani**</documentation></annotation></enumeration>  <enumeration value=**"QAR"**><annotation><documentation>**Qatar Riyal**</documentation></annotation></enumeration>  <enumeration value=**"RON"**><annotation><documentation>**Romania Leu**</documentation></annotation></enumeration>  <enumeration value=**"RSD"**><annotation><documentation>**Serbia Dinar**</documentation></annotation></enumeration>  <enumeration value=**"RUB"**><annotation><documentation>**Russia Ruble**</documentation></annotation></enumeration>  <enumeration value=**"RWF"**><annotation><documentation>**Rwanda Franc**</documentation></annotation></enumeration>  <enumeration value=**"SAR"**><annotation><documentation>**Saudi Arabia Riyal**</documentation></annotation></enumeration>  <enumeration value=**"SBD"**><annotation><documentation>**Solomon Islands Dollar**</documentation></annotation></enumeration>  <enumeration value=**"SCR"**><annotation><documentation>**Seychelles Rupee**</documentation></annotation></enumeration>  <enumeration value=**"SDG"**><annotation><documentation>**Sudan Pound**</documentation></annotation></enumeration>  <enumeration value=**"SEK"**><annotation><documentation>**Sweden Krona**</documentation></annotation></enumeration>  <enumeration value=**"SGD"**><annotation><documentation>**Singapore Dollar**</documentation></annotation></enumeration>  <enumeration value=**"SHP"**><annotation><documentation>**Saint Helena Pound**</documentation></annotation></enumeration>  <enumeration value=**"SLL"**><annotation><documentation>**Sierra Leone Leone**</documentation></annotation></enumeration>  <enumeration value=**"SOS"**><annotation><documentation>**Somalia Shilling**</documentation></annotation></enumeration>  <enumeration value=**"SPL"**><annotation><documentation>**Seborga Luigino**</documentation></annotation></enumeration>  <enumeration value=**"SRD"**><annotation><documentation>**Suriname Dollar**</documentation></annotation></enumeration>  <enumeration value=**"STN"**><annotation><documentation>**Sao Tome and Principe Dobra**</documentation></annotation></enumeration>  <enumeration value=**"SVC"**><annotation><documentation>**El Salvador Colon**</documentation></annotation></enumeration>  <enumeration value=**"SYP"**><annotation><documentation>**Syria Pound**</documentation></annotation></enumeration>  <enumeration value=**"SZL"**><annotation><documentation>**eSwatini Lilangeni**</documentation></annotation></enumeration>  <enumeration value=**"THB"**><annotation><documentation>**Thailand Baht**</documentation></annotation></enumeration>  <enumeration value=**"TJS"**><annotation><documentation>**Tajikistan Somoni**</documentation></annotation></enumeration>  <enumeration value=**"TMT"**><annotation><documentation>**Turkmenistan Manat**</documentation></annotation></enumeration>  <enumeration value=**"TND"**><annotation><documentation>**Tunisia Dinar**</documentation></annotation></enumeration>  <enumeration value=**"TOP"**><annotation><documentation>**Tonga Pa'anga**</documentation></annotation></enumeration>  <enumeration value=**"TRY"**><annotation><documentation>**Turkey Lira**</documentation></annotation></enumeration>  <enumeration value=**"TTD"**><annotation><documentation>**Trinidad and Tobago Dollar**</documentation></annotation></enumeration>  <enumeration value=**"TVD"**><annotation><documentation>**Tuvalu Dollar**</documentation></annotation></enumeration>  <enumeration value=**"TWD"**><annotation><documentation>**Taiwan New Dollar**</documentation></annotation></enumeration>  <enumeration value=**"TZS"**><annotation><documentation>**Tanzania Shilling**</documentation></annotation></enumeration>  <enumeration value=**"UAH"**><annotation><documentation>**Ukraine Hryvnia**</documentation></annotation></enumeration>  <enumeration value=**"UGX"**><annotation><documentation>**Uganda Shilling**</documentation></annotation></enumeration>  <enumeration value=**"USD"**><annotation><documentation>**United States Dollar**</documentation></annotation></enumeration>  <enumeration value=**"UYU"**><annotation><documentation>**Uruguay Peso**</documentation></annotation></enumeration>  <enumeration value=**"UZS"**><annotation><documentation>**Uzbekistan Som**</documentation></annotation></enumeration>  <enumeration value=**"VEF"**><annotation><documentation>**Venezuela Bolivar**</documentation></annotation></enumeration>  <enumeration value=**"VND"**><annotation><documentation>**Viet Nam Dong**</documentation></annotation></enumeration>  <enumeration value=**"VUV"**><annotation><documentation>**Vanuatu Vatu**</documentation></annotation></enumeration>  <enumeration value=**"WST"**><annotation><documentation>**Samoa Tala**</documentation></annotation></enumeration>  <enumeration value=**"XAF"**><annotation><documentation>**Communaute Financiere Africaine (BEAC) CFA Franc BEAC**</documentation></annotation></enumeration>  <enumeration value=**"XCD"**><annotation><documentation>**East Caribbean Dollar**</documentation></annotation></enumeration>  <enumeration value=**"XDR"**><annotation><documentation>**International Monetary Fund (IMF) Special Drawing Rights**</documentation></annotation></enumeration>  <enumeration value=**"XOF"**><annotation><documentation>**Communaute Financiere Africaine (BCEAO) Franc**</documentation></annotation></enumeration>  <enumeration value=**"XPF"**><annotation><documentation>**Comptoirs Francais du Pacifique (CFP) Franc**</documentation></annotation></enumeration>  <enumeration value=**"YER"**><annotation><documentation>**Yemen Rial**</documentation></annotation></enumeration>  <enumeration value=**"ZAR"**><annotation><documentation>**South Africa Rand**</documentation></annotation></enumeration>  <enumeration value=**"ZMW"**><annotation><documentation>**Zambia Kwacha**</documentation></annotation></enumeration>  <enumeration value=**"ZWD"**><annotation><documentation>**Zimbabwe Dollar**</documentation></annotation></enumeration>  </restriction>  </simpleType>  </schema> |

1. *yyyy*- godina; *MM*-mjesec; *dd*-dan; *HH*-sati; *mm* – minuta; *SS* -sekunde; *ENU Code* – kôd ENU; *IKOF*-jedinstveni identifikacioni kôd; *request* – zahtjev. [↑](#footnote-ref-2)