Einvoice Service

(Version v01)

Technical Specification

Versions

Version	Description of Change
v01	Initial Version

Related Documents

Document Name	Description
Einvoice Service - Functional specification	All processes described.
European standard EN 16931	European Standard on eInvoicing. Can downloaded from EN 16931

CONTENTS

1. Introduction

This document provides a description of the data interface for exchanging einvoices between two taxpayers using Central Information System from Tax Administration. The document is intended for all interested parties that will participate in the implementation of the solution on the side of the taxpayer.

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 0 and 8 to this document.

This document provides specification for the einvoice 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 GDT website.

1.1 USED ABBREVIATIONS

Abbreviation	Description	Terminology used in the Law (if it is different)
CA	Certificate Authority	-
CIS	Central Invoice System	-
CRL	Certificate Revocation List	-
FIC	Fiscal Identification Code (generated at server side after successful verification of the invoice)	UII – Unique invoice identifier
EIC	Einvoice Identification Code (generated at server side after successful verification of the einvoice)	-
FWTNIC	Fiscal WTN Identification Code (generated at server side after successful verification of the warehouse transfer note)	UWTNI - Unique Warehouse Transfer Note Identifier
GUID	Global Unique Identifier	-
IIC	Invoice Identification Code	ISC - Invoice Issuer's Security Code
WTNIC	WTN Identification Code (warehouse transfer note identification code)	WTNISC - Warehouse Transfer Note Issuer's Security Code
NUIS	National Unique Identification Number	NUIS/NIPT
TIN	Taxpayer Identification Number	-
OCSP	On-Line Certificate Status Protocol	-
SOAP	Message exchange protocol for XML messages as specified at: https://www.w3.org/TR/soap/	-
ERP	Enterprise Resource Planning (Software application used to interact with CIS einvoice service)	-
UC	Use case	-
UUID	Universally Unique Identifier	-
WSDL	Web Services Description Language –XML-based language for description of functions offered by a WWW service as specified at http://www.w3.org/TR/wsdl	-
XML Schema	A XML-based language intended for definition of XML document structure as specified at http://www.w3.org/TR/xmlschema11-1/ and https://www.w3.org/TR/xmlschema11-2/	-

Table 1 – Used abbreviations

1.2 TERMINOLOGY

Term	Definition	Terminology used in the Law (if it is different)
Response data message	A data structure in a defined format prescribed by the financial authority, which contains the Einvoice Identification Code (EIC) and is used as acknowledgement of invoice and formal correctness of the registered invoice data message sent.	-
Error Data Message	A data structure in a defined format prescribed by the financial authority, which contains an error code and its text description as a reaction to a registered invoice data message received containing critical errors preventing it from being processed, or when another error occurs which prevents the message being processed at the tax authority's side.	-
Invoice	An einvoice is a proof of electronically sale issued by a taxpayer to a person or entity making a purchase, which contains all information regarding totals of the sale and items.	-
Issuer	Person who is issuing the einvoice. Issuer of the einvoice is responsible for the fiscalization and generating einvoice in CIS. This person is in most cases the seller of goods and services but in case of self-billing invoice, the issuer is the buyer of goods and services.	-
Registered Einvoice	Einvoice which is registered on CIS containing EIC.	-
Registered einvoice data message	A data structure in a defined format prescribed by the fiscal authority, which contains information about the sale and other technical information necessary. This is a complete XML message containing information described in the relevant Web service standards: SOAP/WSDL/WS-Security, etc. A registered invoice data message is sent by an ERP to the tax authority's common technical equipment (Central invoice system).	-
Central invoice platform	Central invoice platform is a web application for taxpayers providing support for einvoice processes.	-
Schematron	Set of rules defined by the "European standard EN 16931" specification. Rules are used to check whether provided XML is compliant with the specification.	-

Table 2 - Terminology

2. 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 solely by software developers (developing ERP software), not by end users. Sending a data message to the non-production environment shall not be considered sending of registered einvoice information. The EIC returned by the non-production environment is not a valid EIC (it is different per prefix). In the non-production environment, digital certificates for cash registers may be issued using a simplified process.
- **Production environment** is intended for the taxpayers and will be used for routine operations.

Endpoints:

- Test environment:
  - https://einvoice-test.tatime.gov.al/EinvoiceService-v1/EinvoiceService.wsdl

2.1 PREPARATION WORKS FOR EINVOICE SERVICE USE

Details on this matter can be found in “Einvoice Service - Functional specification”, chapter that covers this subject.

2.2 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.

2.2.1 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.

2.2.2 INTERNET CONNECTION

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

2.2.3 INFORMATION SYSTEM OF THE CLIENT

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.

2.3 CONDITIONS FOR CONNECTION TO 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 NAIS. Environments are not different in its functionality (besides new functionalities development), only difference is in data – test environment uses test data.

2.3.1 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**

Network recommendations for client system are:

Link characteristics	**Permanent symmetric link**
Bandwidth	**2 Mb/s at minimum (up to 40 messages per second with assumption that message takes 6 KB)**

2.3.2 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 NAIS production CA, while in test environment the certificate is issued by NAIS test CA.

Protection at the transport layer	**HTTPS (TLS v1.1 and v1.2, AES_256 encryption at least)**
Certificates for the electronic signing	**Certificate type: application digital certificate for fiscalization and einvoice issuing**

2.3.3 APLICATION 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**

3. Interface

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

The web service has several operations for manipulating einvoices by ERP software. Each of 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 4.3.

3.1 INTERFACE VERSIONING

Versioning of the einvoice 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. /EinvoiceService-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.

3.2 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 einvoice 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.

3.3 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 einvoice service respectably. Exception to that rule are error messages (described in the chapter 3.6) which are not signed by the einvoice 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.

3.4 CONTROLS

There are three types of controls:

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

Mandatory controls shall be performed by CIP 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
Check size of the data	Size must not exceed 2GB	1
XML structure validation	Check of the individual registered invoice data message's in XML against the XSD schema (*.xsd). XSD schema contains an exact definition of the data and format structure for the individual data items and a check of presence of individual items	11
Certificate validation	Check that certificate is not expired. Check that the certificate is issued by the trusting CA. Check that the identification number in the certificate corresponds to the invoice issuer identification number (tax number) in the XML message. Check that certificate is not listed in CRL.	34, 35, 36, 37, 38, 39
Electronic signature check	Check that the hash of the message calculated by CIS corresponds to the hash listed in the message. Check that the signature corresponds to the hash of the message and to the public key of the certificate.	22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34
Client time differs	Client time differs from a server's time by more than allowed time in minutes	2
Other controls	Other controls as specified in the functional specification and for control purposes of the Tax Administration	

Table 3

3.5 REGISTER EINVOICE

ERP of the einvoice Issuer uses this method to deliver einvoices to the CIS. Actual einvoice is contained inside delivered request message as data specified by standards ISO/IEC 19845: 2015 (UBL 2.1) and UN/CEFACT Interindustrial Invoice (XML 16B schemes) described in "European standard EN 16931" specification.

Data exchange process starts when the issuer is about to issue an einvoice to the customer. Before einvoice is created, ERP must retrieve FIC from fiscalization service and incorporate it inside an einvoce. Details on mapping of fields between einvoice and fiscalization message can found in the annex 6. Einvoice is then signed and incorporated inside request data message which is again signed with same private key. After that the 1-way TLS communication is started and once successful it calls the service.

CIS receives and processes request message. If the request is successfully processed, CIS creates signed PDF representing einvoice and sends back to the ERP XML message that contains EIC, which is unique for every einvoice.

ERP receives answer message and checks its electronic signature.

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 EIC.

Detail description of the process and rules can be found in “Einvoice Service - Functional specification”, chapter that covers this subject.

3.5.1 REGISTER EINVOICE REQUEST DATA MESSAGE

Name	Field type	Occurrence [Min, Max]	Description
RegisterEinvoiceRequest	Root	[1, 1]	Root XML element representing register einvoice message.
Id	Attribute	[1, 1]	Attribute used for signature creation and verification. Fixed value “Request”.
Version	Attribute	[1,1]	Attribute used to specify compliance with XSD schema. For this version fixed value is “1”.
Header	Element	[1, 1]	XML element representing header of the invoice containing data about the message (request) sent.
UUID	Attribute	[1, 1]	UUID generated by an ERP for every new message send to the CIS.
SendDateTime	Attribute	[1, 1]	Date and time of sending the message from an ERP to the CIS.
EinvoiceEnvelope	Element	[1, 1]	XML element containing one einvoice.
UblInvoice	Element	[0, 1]	XML element containing UBL invoice XML encoded as Base64 string.
UblCreditNote	Element	[0, 1]	XML element containing UBL credit note XML encoded as Base64 string.
CiiInvoice	Element	[0, 1]	XML element containing CII invoice XML encoded as Base64 string.
Signature	Element	[1, 1]	XML element representing signature for the message.

Table 4

3.5.1.1 Header

XML element representing header of the request data message.

3.5.1.2 Header UUID

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

Data type	string
Length	36 characters
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

3.5.1.3 Header SendDateTime

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}`
Example	2019-01-24T22:00:58+01:00 2019-01-24T22:00:58-01:00

Table 5

3.5.1.4 EinvoiceEnvelope

XML element containing one einvoice data element.

3.5.1.5 EinvoiceEnvelope UblInvoice

XML element representing einvoice invoice type created as per ISO/IEC 19845: 2015 (UBL 2.1) standard described in “European standard EN 16931” specification. Einvoice is base64 encoded.

Data type	base64Binary
Example	PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0idXRmLTgiIHN0YW5kYWxvbmU9InllcyI/.........pbmU+DQo8L0ludm9pY2U+

Table 6

3.5.1.6 EinvoiceEnvelope UblCreditNote

XML element representing einvoice credit note type created as per ISO/IEC 19845: 2015 (UBL 2.1) standard described in “European standard EN 16931” specification. Einvoice is base64 encoded.

Data type	base64Binary
Example	PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0idXRmLTgiIHN0YW5kYWxvbmU9InllcyI/.........pbmU+DQo8L0ludm9pY2U+

Table 7

3.5.1.7 EinvoiceEnvelope CiiInvoice

XML element representing einvoice invoice type created as per UN/CEFACT Interindustrial Invoice (XML 16B schemes) standard described in “European standard EN 16931” specification. Einvoice is base64 encoded.

Data type	base64Binary
Example	PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0idXRmLTgiIHN0YW5kYWxvbmU9InllcyI/.........pbmU+DQo8L0ludm9pY2U+

Table 8

3.5.1.8 Signature

XML element stores enveloped digital signature described in the chapter 4.3.1.

3.5.2 REGISTER EINVOICE RESPONSE DATA MESSAGE

Name	Field type	Occurrence [Min, Max]	Description
RegisterInvoiceResponse	Element	[1, 1]	Root XML element representing register einvoice response message.
Id	Attribute	[1, 1]	Attribute used for signature creation and verification. Fixed value “Response”.
Version	Attribute	[1,1]	Attribute used to specify compliance with XSD schema. For this version fixed value is “1”.
Header	Element	[1, 1]	XML element representing generic message data about the response sent.
UUID	Attribute	[1, 1]	UUID generated by a CIS for every register invoice response data message send to the ERP.
RequestUUID	Attribute	[1, 1]	UUID of the request message for which this response message was sent.
SendDateTime	Attribute	[1, 1]	Date and time of sending the register einvoice response data message from a CIS to the ERP.
EIC	Element	[1, 1]	CIS generated verification code that can be used to uniquely identify registered einvoice.

	Warnings			Element	[0, 1]	XML element representing list of warnings.
		Warning		Element	[1, 200]	XML element representing a warning in list of warnings.
			Code	Attribute	[1, 1]	Warning code.
			Description	Attribute	[1, 1]	Warning description.
	Signature			Element	[1, 1]	XML element with signature.

Table 9

3.5.2.1 Header

XML element representing header of the response data message.

3.5.2.2 Header UUID

Element generated by the CIS for every message sent to the ERP. It uniquely identifies the message sent to the ERP. UUID is constructed according to the RFC4122 version 4.

Data type	string
Length	36 characters
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 10

3.5.2.3 Header RequestUUID

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

Data type	string
Length	36 characters
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 11

3.5.2.4 Header SendDateTime

Element represents date and time of sending the response message to the ERP. 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}`
Example	2019-01-24T22:00:58+01:00 2019-01-24T22:00:58-01:00

Table 12

3.5.2.5 EIC

Element represents unique number generated by the CIS under which the requested einvoice is registered.

Data type	string
Length	36 characters
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 13

3.5.2.6 Warnings

XML element that contains list of warnings.

3.5.2.7 Warnings Warning

XML element that contains one warning in the list of warnings.

3.5.2.8 Warnings Warning Code

Code representing a warning. Current implementation only supports schematron validation codes.

Data type	String
Length	20 characters
Example	UBL-CR-001

Table 14

3.5.2.9 Warnings Warning Description

Description of the warning message.

Data type	string
Length	100 characters
Example	[UBL-CR-001]-A UBL invoice should not include extensions

Table 15

3.5.2.10 Signature

XML element stores enveloped digital signature described in the chapter 4.3.1.

3.5.3 MANDATORY CONTROLS

Mandatory controls shall be performed on received data messages in the CIS in real time. When any of the critical controls return a failure, the data message shall not be approved, and valid response 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 3.6). When errors which the system can interpret as a cyber-attack are identified, the system does not send any response to the client (the ERP).

The mandatory controls include all the controls from chapter 3.1 and the following:

Control Name	Control Description (Error if)	Error code
Issuer does not exist	Issuer does not exist.	52
Issuer not active	Issuer is not active in the RTP.	55
Certificate not the same	Private key used to sign request data message is not the same as private key used to sign einvoice data.	40
TIN not the same	TIN inside request certificate is not the same as TIN inside einvoice data.	54

Schematron validation failed	Schematron validations failed.	12
Einvoice extraction	Einvoice not supported. Einvoice extraction failed Einvoice is missing	50, 51, 53

Table 16

3.5.4 ERROR MESSAGE

Error message is defined in chapter 3.6.

3.5.5 EXAMPLE XML

3.5.5.1 Request XML

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <RegisterEinvoiceRequest xmlns="https://Einvoice.tatime.gov.al/EinvoiceService/schema" xmlns:ns2="http://www.w3.org/2000/09/xmldsig#"
Id="Request" Version="3">
            <Header SendDateTime="2020-09-10T15:45:24+02:00" UUID="e7f25589-3882-4270-ae0c-05d2da18fb73"/>
            <EinvoiceEnvelope>
                <UblInv>PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0idXRmLTgiIHN0Y.........C9jYWM6SW52b2ljZUxpbmU+DQo8L0ludm9pY2U+</UblInv>
            </EinvoiceEnvelope>
            <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>0WVxH5gC1P.........JgtU+rADq9m7kQ4=</DigestValue>
                    </Reference>
                </SignedInfo>
                <SignatureValue>BPIZIjKshHTDVe.........ADImOdRhdG7t9jQ==</SignatureValue>
                <KeyInfo>
                    <X509Data>
                        <X509Certificate>MIIFCDCCAvCgAw.........XdhRXasm+AZhWTGovAAqeLvDyjpAg=</X509Certificate>
                    </X509Data>
                </KeyInfo>
            </Signature>
        </RegisterEinvoiceRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.5.5.2 Response XML

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <SOAP-ENV:Body>
      <ns2:RegisterEinvoiceResponse Id="Response" Version="1" xmlns:ns2="https://Einvoice.tatime.gov.al/EinvoiceService/schema"
xmlns:ns3="http://www.w3.org/2000/09/xmldsig#">
         <ns2:Header RequestUUID="e7f25589-3882-4270-ae0c-05d2da18fb73" SendDateTime="2020-09-14T15:02:57Z" UUID="3799b13d-421b-44a5-a874-
6c3c063e0df0"/>
         <ns2:EIC>5e8d2330-8495-43ce-aa33-5181f0a1eac6</ns2:EIC>
         <ns2:Warnings>
            <ns2:Warning Code="UBL-CR-001" Text="[UBL-CR-001]-A UBL invoice should not include extensions"/>
            <ns2:Warning Code="UBL-CR-006" Text="[UBL-CR-006]-A UBL invoice should not include the IssueTime "/>
         </ns2:Warnings>
         <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>3Cep3Ap4EEZ/.........AhosJtELsXF0=</DigestValue>
               </Reference>
            </SignedInfo>
            <SignatureValue>VJBiOpoSjZTqJ3NXNw+Os.........7ieoyJP9hTnKA==</SignatureValue>
            <KeyInfo>
               <X509Data>
                  <X509Certificate>MIIFRzCCBC+gAwIBAgI.........4RjtKkBTwYahJEZb8DE3jc9BSYwpAaAE=</X509Certificate>
               </X509Data>
            </KeyInfo>
         </Signature>
```

```
        </ns2:RegisterEinvoiceResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.6 CHANGE EINVOICE STATUS

ERP of the issuer can use this method to change status of exiting einvoice.

3.6.1 CHANGE EINVOICE STATUS REQUEST DATA MESSAGE

Name	Field type	Occurrence [Min, Max]	Description
EinvoiceChangeStatusRequest	Root	[1, 1]	Root XML element representing request message.
Id	Attribute	[1, 1]	Attribute used for signature creation and verification. Fixed value “Request”.
Version	Attribute	[1,1]	Attribute used to specify compliance with XSD schema. For this version fixed value is “1”.
Header	Element	[1, 1]	XML element representing header of the invoice containing data about the message (request) sent.
UUID	Attribute	[1, 1]	UUID generated by an ERP for every new message send to the CIS.
SendDateTime	Attribute	[1, 1]	Date and time of sending the message from an ERP to the CIS.
EICs	Element	[1, 1]	XML element containing list of EIC for status change.
EIC	Element	[1, 100]	EIC for status change.
EinStatus	Element	[1, 1]	New desired status of einvoices in the EICs list.
Signature	Element	[1, 1]	XML element representing signature for the message.

Table 17

3.6.1.1 Header

XML element representing header of the request data message.

3.6.1.2 Header UUID

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

Data type	string
Length	36 characters
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

3.6.1.3 Header SendDateTime

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}`
Example	2019-01-24T22:00:58+01:00 2019-01-24T22:00:58-01:00

Table 18

3.6.1.4 EICs

XML element that contains list of EIC for which is requested status change.

3.6.1.5 EICs EIC

XML element that contains one EIC in the list of EICs for status change.

Data type	string
Length	36 characters
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 19

3.6.1.6 EinStatus

This element represents new desired status of einvoices in the EICs

Data type	string
Values	Enumeration, described in the table below.
Example	ACCEPTED

Table 20

Following table shows the list of allowed values.

Value	Description
ACCEPTED	Accepted
REFUSED	Refused
PAID	Paid
PARTIALLY_PAID	Partially paid

Table 21

3.6.2 CHANGE EINVOICE STATUS RESPONSE DATA MESSAGE

Name	Field type	Occurrence [Min, Max]	Description
EinvoiceChangeStatusResponse	Element	[1, 1]	Root XML element representing response message.
Id	Attribute	[1, 1]	Attribute used for signature creation and verification. Fixed value “Response”.
Version	Attribute	[1,1]	Attribute used to specify compliance with XSD schema. For this version fixed value is “1”.
Header	Element	[1, 1]	XML element representing generic message data about the response sent.
UUID	Attribute	[1, 1]	UUID generated by a CIS for every new response data message send to the ERP.
RequestUUID	Attribute	[1, 1]	UUID of the request message for which this response message was sent.
SendDateTime	Attribute	[1, 1]	Date and time of sending the response data message from a CIS to the ERP.
ResponseCode	Element	[1, 1]	XML element representing response code.
Signature	Element	[1, 1]	XML element with signature.

Table 22

3.6.2.1 Header

XML element representing header of the response data message.

3.6.2.2 Header UUID

Element generated by the CIS for every message sent to the ERP. It uniquely identifies the message sent to the ERP. UUID is constructed according to the RFC4122 version 4.

Data type	string
Length	36 characters
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 23

3.6.2.3 Header RequestUUID

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

Data type	string
Length	36 characters
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 24

3.6.2.4 Header SendDateTime

Element represents date and time of sending the response message to the ERP. 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}`
Example	2019-01-24T22:00:58+01:00 2019-01-24T22:00:58-01:00

Table 25

3.6.2.5 ResponseCode

This element represents status of the status change. If all einvoices are changed then response is equal to ACCEPTED, otherwise Error message is returned.

Data type	string
Values	Enumeration, described in the table below.
Example	ACCEPTED

Table 26

Following table shows the list of allowed values.

Value	Description
ACCEPTED	Accepted

Table 27

3.6.3 MANDATORY CONTROLS

Mandatory controls shall be performed on received data messages in the CIS in real time. When any of the critical controls return a failure, the data message shall not be approved, and valid response 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 3.6). When errors which the system can interpret as a cyber-attack are identified, the system does not send any response to the client (the ERP).

The mandatory controls include all the controls from chapter 3.1 and the following:

Control Name	Control Description (Error if)	Error code
Issuer does not exist	Issuer does not exist.	52
Issuer not active	Issuer is not active in the RTP.	55

Table 28

3.6.4 ERROR MESSAGE

Error message is defined in chapter 3.6.

3.6.5 EXAMPLE XML

3.6.5.1 Request XML

```xml
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <EinvoiceChangeStatusRequest xmlns="https://Einvoice.tatime.gov.al/EinvoiceService/schema"
xmlns:ns2="http://www.w3.org/2000/09/xmldsig#" Id="Request" Version="3">
            <Header SendDateTime="2020-09-10T15:45:24+02:00" UUID="e7f25589-3882-4270-ae0c-05d2da18fb73"/>
            <EICs>
                <EIC>e8e25589-3882-4270-ae0c-05d2da18fb73</EIC>
                <EIC>a6e25467-2771-4280-bc0c-12d3da18fb73</EIC>
            </EICs>
            <EinStatus>ACCEPTED</EinStatus>
            <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>0WVxH5gC1P.........JgtU+rADq9m7kQ4=</DigestValue>
                    </Reference>
                </SignedInfo>
                <SignatureValue>BPIZIjKshHTDVe.........ADImOdRhdG7t9jQ==</SignatureValue>
                <KeyInfo>
                    <X509Data>
                        <X509Certificate>MIIFCDCCAvCgAw.........XdhRXasm+AZhWTGovAAqeLvDyjpAg=</X509Certificate>
                    </X509Data>
                </KeyInfo>
            </Signature>
        </EinvoiceChangeStatusRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.6.5.2 Response XML

```xml
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <SOAP-ENV:Body>
      <ns2:EinvoiceChangeStatusResponse Id="Response" Version="1" xmlns:ns2="https://Einvoice.tatime.gov.al/EinvoiceService/schema"
xmlns:ns3="http://www.w3.org/2000/09/xmldsig#">
         <ns2:Header RequestUUID="e7f25589-3882-4270-ae0c-05d2da18fb73" SendDateTime="2020-09-14T15:02:57Z" UUID="3799b13d-421b-44a5-a874-
6c3c063e0df0"/>
         <ResponseCode>ACCEPTED</ResponseCode>
         <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>3Cep3Ap4EEZ/.........AhosJtELsXF0=</DigestValue>
               </Reference>
            </SignedInfo>
            <SignatureValue>VJBiOpoSjZTqJ3NXNw+Os.........7ieoyJP9hTnKA==</SignatureValue>
            <KeyInfo>
               <X509Data>
                  <X509Certificate>MIIFRzCCBC+gAwIBAgI..........4RjtKkBTwYahJEZb8DE3jc9BSYwpAaAE=</X509Certificate>
               </X509Data>
            </KeyInfo>
         </Signature>
      </ns2:EinvoiceChangeStatusResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.7 GET EINVOICES

ERP of the issuer can use this method to retrieve his einvoices. If EIC of the einvoice is specified than PDF of that einvoice is also retrieved.

3.7.1 GET EINVOICES REQUEST DATA MESSAGE

Name	Field type	Occurrence [Min, Max]	Description
GetEinvoicesRequest	Root	[1, 1]	Root XML element representing response message.
Id	Attribute	[1, 1]	Attribute used for signature creation and verification. Fixed value “Request”.
Version	Attribute	[1,1]	Attribute used to specify compliance with XSD schema. For this version fixed value is “1”.
Header	Element	[1, 1]	XML element representing header of the invoice containing data about the message (request) sent.
UUID	Attribute	[1, 1]	UUID generated by an ERP for every new message send to the CIS.
SendDateTime	Attribute	[1, 1]	Date and time of sending the message from an ERP to the CIS.
EIC	Element	[0, 1]	EIC value for find one einvoice. If set, this method only finds one einvoice and returns created PDF representing that einvoice.
PartyType	Element	[0, 1]	Einvoice party type (Buyer or Seller).
RecDateTimeFrom	Element	[0, 1]	Receive date time from.
RecDateTimeTo	Element	[0, 1]	Receive date time to.
Signature	Element	[1, 1]	XML element representing signature for the message.

Table 29

3.7.1.1 Header

XML element representing header of the request data message.

3.7.1.2 Header UUID

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

Data type	string
Length	36 characters
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

3.7.1.3 Header SendDateTime

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}`
Example	2019-01-24T22:00:58+01:00 2019-01-24T22:00:58-01:00

Table 30

3.7.1.4 EIC

With this element ERP can retrieve single einvoice and for that einvoice PDF will be returned in the response.

Data type	string
Length	36 characters
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 31

3.7.1.5 PartyType

With this element ERP can filter einvoices whether taxpayer is buyer or seller of the einvoice.

Data type	string
Values	Enumeration, described in the table below.
Example	BUYER

Table 32

Following table shows the list of allowed values.

Value	Description
BUYER	Buyer
SELLER	Seller

Table 33

3.7.1.6 RecDateTimeFrom

Whit this element ERP can filter einvoices by received date time from period.

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}`
Example	2019-01-24T22:00:58+01:00 2019-01-24T22:00:58-01:00

Table 34

3.7.1.7 RecDateTimeTo

Whit this element ERP can filter einvoices by received date time to period.

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}`
Example	2019-01-24T22:00:58+01:00 2019-01-24T22:00:58-01:00

Table 35

3.7.2 GET EINVOICES RESPONSE DATA MESSAGE

Name	Field type	Occurrence [Min, Max]	Description
GetTaxpayersResponse	Element	[1, 1]	Root XML element representing response message.
Id	Attribute	[1, 1]	Attribute used for signature creation and verification. Fixed value “Response”.

	Version			Attribute	[1,1]	Attribute used to specify compliance with XSD schema. For this version fixed value is “1”.
	Header			Element	[1, 1]	XML element representing generic message data about the response sent.
		UUID		Attribute	[1, 1]	UUID generated by a CIS for every new response data message send to the ERP.
		RequestUUID		Attribute	[1, 1]	UUID of the request message for which this response message was sent.
		SendDateTime		Attribute	[1, 1]	Date and time of sending the response data message from a CIS to the ERP.
	Einvoices			Element	[1, 1]	XML element representing list of einvoices.
		Einvoice		Element	[0, 100]	XML element representing a one einvoice in list of einvoices..
			Pdf	Element	[0, 1]	Signed PDF representation of the einvoice
			EIC	Attribute	[1, 1]	Einvoice EIC.
			DocNumber	Attribute	[1, 1]	Document number
			DocType	Attribute	[1, 1]	Document type
			RecDateTime	Attribute	[1, 1]	Received date time.
			DueDateTime	Attribute	[1, 1]	Due date time
			Status	Attribute	[1, 1]	Status of the einvoice
			Amount	Attribute	[1, 1]	Amount stated in the einvoice
			PartyType	Attribute	[1, 1]	Party type of taxpayer in the einvoice.
	Signature			Element	[1, 1]	XML element with signature.

Table 36

3.7.2.1 Header

XML element representing header of the response data message.

3.7.2.2 Header UUID

Element generated by the CIS for every message sent to the ERP. It uniquely identifies the message sent to the ERP. UUID is constructed according to the RFC4122 version 4.

Data type	string
Length	36 characters
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 37

3.7.2.3 Header RequestUUID

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

Data type	string
Length	36 characters
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

3.7.2.4 Header SendDateTime

Element represents date and time of sending the response message to the ERP. 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}
Example	2019-01-24T22:00:58+01:00 2019-01-24T22:00:58-01:00

Table 39

3.7.2.5 Einvoices

XML element that contains list of einvoices.

3.7.2.6 Einvoices Einvoice

XML element that contains one einvoice in the list of einvoices.

3.7.2.7 Einvoices Einvoice Pdf

XML element that contains signed PDF with einvoice representation.

Data type	base64Binary
Example	PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0idXRmLTgiIHN0YW5kYWxvbmU9InllcyI/……….pbmU+DQo8L0ludm9pY2U+

Table 40

3.7.2.8 Einvoices Einvoice EIC

Attribute represents unique number generated by the CIS under which the requested einvoice is registered.

Data type	string
Length	36 characters
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

3.7.2.9 Einvoices Einvoice DocNumber

Document number as stated in the einvoice.

Data type	string
Length	100 characters
Example	123-2020

Table 42

3.7.2.10 Einvoices Einvoice DocType

Document type as stated in the einvoice.

Data type	string
Values	Enumeration, described in the table below.
Example	380

Table 43

Following table shows the list of allowed values.

Value	Description

80	Debit note related to goods or services type.
82	Metered services invoice type.
84	Debit note related to financial adjustments type.
380	Commercial invoicetype.
383	Debit note type.
386	Prepayment invoice type.
393	Factored invoice type.
395	Consignment invoice type.
575	Insurer's invoice type.
623	Forwarder's invoice type.
780	Freight invoice type.

Table 44

3.7.2.11 Einvoices Einvoice RecDateTime

Received data time of the einvoice in the CIS.

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}`
Example	2019-01-24T22:00:58+01:00 2019-01-24T22:00:58-01:00

Table 45

3.7.2.12 Einvoices Einvoice DueDateTime

Due date time as stated in the einvoice.

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}`
Example	2019-01-24T22:00:58+01:00 2019-01-24T22:00:58-01:00

Table 46

3.7.2.13 Einvoices Einvoice EinStatus

Status of the einvoice inside CIS.

Data type	string
Values	Enumeration, described in the table below.
Example	ACCEPTED

Table 47

Following table shows the list of allowed values.

Value	Description
ACCEPTED	Accepted
REFUSED	Refused
PAID	Paid
PARTIALLY_PAID	Partially paid

Table 48

3.7.2.14 Einvoices Einvoice Amount

Amount as stated in the einvoice.

<table>
<tr><td>Data type</td><td>decimal</td></tr>
<tr><td>Pattern</td><td><code>-?([1-9][0-9]*|0)\.[0-9]{2}|0</code></td></tr>
<tr><td>Example</td><td>17.24</td></tr>
</table>

Table 49

3.7.2.15 Einvoices Einvoice PartyType

Type of the issuer in the einvoice.

Data type	string
Values	Enumeration, described in the table below.
Example	BUYER

Table 50

Following table shows the list of allowed values.

Value	Description
BUYER	Buyer
SELLER	Seller

Table 51

3.7.3 MANDATORY CONTROLS

Mandatory controls shall be performed on received data messages in the CIS in real time. When any of the critical controls return a failure, the data message shall not be approved, and valid response 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 3.6). When errors which the system can interpret as a cyber-attack are identified, the system does not send any response to the client (the ERP).

The mandatory controls include all the controls from chapter 3.1 and the following:

Control Name	Control Description (Error if)	Error code
Issuer does not exist	Issuer does not exist.	52
Issuer not active	Issuer is not active in the RTP.	55

Table 52

3.7.4 ERROR MESSAGE

Error message is defined in chapter 3.6.

3.7.5 EXAMPLE XML

3.7.5.1 Request XML

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <GetTaxpayersRequest xmlns="https://Einvoice.tatime.gov.al/EinvoiceService/schema" xmlns:ns2="http://www.w3.org/2000/09/xmldsig#"
Id="Request" Version="3">
```

```
            <Header SendDateTime="2020-09-10T15:45:24+02:00" UUID="e7f25589-3882-4270-ae0c-05d2da18fb73"/>
            <EIC>e8e25589-3882-4270-ae0c-05d2da18fb73</EIC>
            <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>0WVxH5gC1P.........JgtU+rADq9m7kQ4=</DigestValue>
                    </Reference>
                </SignedInfo>
                <SignatureValue>BPIZIjKshHTDVe.........ADImOdRhdG7t9jQ==</SignatureValue>
                <KeyInfo>
                    <X509Data>
                        <X509Certificate>MIIFCDCCAvCgAw.........XdhRXasm+AZhWTGovAAqeLvDyjpAg=</X509Certificate>
                    </X509Data>
                </KeyInfo>
            </Signature>
        </GetTaxpayersRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.7.5.2 Response XML

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <SOAP-ENV:Body>
      <ns2:GetTaxpayersResponse Id="Response" Version="1" xmlns:ns2="https://Einvoice.tatime.gov.al/EinvoiceService/schema"
xmlns:ns3="http://www.w3.org/2000/09/xmldsig#">
         <ns2:Header RequestUUID="e7f25589-3882-4270-ae0c-05d2da18fb73" SendDateTime="2020-09-14T15:02:57Z" UUID="3799b13d-421b-44a5-a874-
6c3c063e0df0"/>
         <Einvoices>
             <Einvoice EIC="e7f25589-3882-4270-ae0c-05d2da18fb73" DocNumber="123-2020" DocType="380" RecDateTime="2020-09-10T15:45:24+02:00"
DueDateTime="2020-09-10T15:45:24+02:00" Status="ACCEPTED" Amount="100.00" PartyType="SELLER"/>
         </Einvoices>
         <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>3Cep3Ap4EEZ/.........AhosJtELsXF0=</DigestValue>
               </Reference>
            </SignedInfo>
            <SignatureValue>VJBiOpoSjZTqJ3NXNw+Os.........7ieoyJP9hTnKA==</SignatureValue>
            <KeyInfo>
               <X509Data>
                  <X509Certificate>MIIFRzCCBC+gAwIBAgI.........4RjtKkBTwYahJEZb8DE3jc9BSYwpAaAE=</X509Certificate>
               </X509Data>
            </KeyInfo>
         </Signature>
      </ns2:GetTaxpayersResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.8 GET TAXPAYERS

ERP of the issuer can use this method to retrieve existing taxpayer from CIS.

3.8.1 GET TAXPAYERS REQUEST DATA MESSAGE

Name	Field type	Occurrence [Min, Max]	Description
GetTaxpayersRequest	Root	[1, 1]	Root XML element representing request message.
Id	Attribute	[1, 1]	Attribute used for signature creation and verification. Fixed value “Request”.
Version	Attribute	[1,1]	Attribute used to specify compliance with XSD schema. For this version fixed value is “1”.
Header	Element	[1, 1]	XML element representing header of the invoice containing data about the message (request) sent.
UUID	Attribute	[1, 1]	UUID generated by an ERP for every new message send to the CIS.
SendDateTime	Attribute	[1, 1]	Date and time of sending the message from an ERP to the CIS.
Filter	Element	[1, 1]	XML element containing choice filter for get taxpayer request message.
Tin	Element	[0, 1]	Element contains filter for Nuis.

		Name	Element	[0, 1]	Element contains filter for name.
	Signature		Element	[1, 1]	XML element representing signature for the message.

Table 53

3.8.1.1 Header

XML element representing header of the request data message.

3.8.1.2 Header UUID

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

Data type	string
Length	36 characters
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

3.8.1.3 Header SendDateTime

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}`
Example	2019-01-24T22:00:58+01:00 2019-01-24T22:00:58-01:00

Table 54

3.8.1.4 Filter

XML element containing choice filter for get taxpayer request message.

3.8.1.5 Filter Tin

Filter by taxpayer NUIS.

Data type	string
Length	10
Pattern	`[a-zA-Z]{1}[0-9]{8}[a-zA-Z]{1}`
Example	K72001008V

Table 55

3.8.1.6 Filter Name

Filter by taxpayer name.

Data type	string
Minimum length	3 characters
Maximum length	100 characters
Example	123-2020

Table 56

3.8.2 GET TAXPAYERS RESPONSE DATA MESSAGE

Name	Field type	Occurrence [Min, Max]	Description
GetEinvoicesResponse	Element	[1, 1]	Root XML element representing response message.
Id	Attribute	[1, 1]	Attribute used for signature creation and verification. Fixed value “Response”.
Version	Attribute	[1,1]	Attribute used to specify compliance with XSD schema. For this version fixed value is “1”.
Header	Element	[1, 1]	XML element representing generic message data about the response sent.
UUID	Attribute	[1, 1]	UUID generated by a CIS for every new response data message send to the ERP.
RequestUUID	Attribute	[1, 1]	UUID of the request message for which this response message was sent.
SendDateTime	Attribute	[1, 1]	Date and time of sending the response data message from a CIS to the ERP.
Taxpayers	Element	[1, 1]	XML element representing list of taxpayers
Taxpayer	Element	[0, 100]	XML element representing a one taxpayer in list of taxpayers..
Name	Attribute	[1, 1]	Name of the taxpayer
Tin	Attribute	[1, 1]	TIN of the taxpayer
Address	Attribute	[1, 1]	Address of the taxpayer
Town	Attribute	[1, 1]	Town of taxpayer
Country	Attribute	[1, 1]	Country of taxpayer
Signature	Element	[1, 1]	XML element with signature.

Table 57

3.8.2.1 Header

XML element representing header of the response data message.

3.8.2.2 Header UUID

Element generated by the CIS for every message sent to the ERP. It uniquely identifies the message sent to the ERP. UUID is constructed according to the RFC4122 version 4.

Data type	string
Length	36 characters
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 58

3.8.2.3 Header RequestUUID

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

Data type	string
Length	36 characters
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 59

3.8.2.4 Header SendDateTime

Element represents date and time of sending the response message to the ERP. 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}`
Example	2019-01-24T22:00:58+01:00 2019-01-24T22:00:58-01:00

Table 60

3.8.2.5 Taxpayers

XML element that contains list of taxpayers.

3.8.2.6 Taxpayers Taxpayer

XML element that contains one taxpayer in the list of taxpayers.

3.8.2.7 Taxpayers Taxpayer Name

Taxpayer’s name.

Data type	string
Length	100 characters
Example	Company name

Table 61

3.8.2.8 Taxpayers Taxpayer Tin

Taxpayer’s NUIS.

Data type	string
Length	10
Pattern	`[a-zA-Z]{1}[0-9]{8}[a-zA-Z]{1}`
Example	K72001008V

Table 62

3.8.2.9 Taxpayers Taxpayer Address

Taxpayer’s address.

Data type	string
Length	100 characters
Example	Rua 1

Table 63

3.8.2.10 Taxpayers Taxpayer Town

Taxpayer’s town.

Data type	string
Length	100 characters

Example	Tirana

Table 64

3.8.2.11 Taxpayers Taxpayer Country

Taxpayer’s country represented as ISO 3166-1 Alpha-3 code.

Data type	string
Constraint	Enumeration, described in the table below.
Example	ALB

Table 65

Enumeration values for selected countries are listed in table below.

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

Table 66

3.8.3 MANDATORY CONTROLS

Mandatory controls shall be performed on received data messages in the CIS in real time. When any of the critical controls return a failure, the data message shall not be approved, and valid response 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 3.6). When errors which the system can interpret as a cyber-attack are identified, the system does not send any response to the client (the ERP).

The mandatory controls include all the controls from chapter 3.1 and the following:

Control Name	Control Description (Error if)	Error code
Issuer does not exist	Issuer does not exist.	52
Issuer not active	Issuer is not active in the RTP.	55

Table 67

3.8.4 ERROR MESSAGE

Error message is defined in chapter 3.6.

3.8.5 EXAMPLE XML

3.8.5.1 Request XML

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <GetTaxpayersRequest xmlns="https://Einvoice.tatime.gov.al/EinvoiceService/schema" xmlns:ns2="http://www.w3.org/2000/09/xmldsig#"
Id="Request" Version="3">
```

```
            <Header SendDateTime="2020-09-10T15:45:24+02:00" UUID="e7f25589-3882-4270-ae0c-05d2da18fb73"/>
            <Filter>
                <Name>Comp</Name>
            </Filter>
            <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>0WVxH5gC1P.........JgtU+rADq9m7kQ4=</DigestValue>
                    </Reference>
                </SignedInfo>
                <SignatureValue>BPIZIjKshHTDVe.........ADImOdRhdG7t9jQ==</SignatureValue>
                <KeyInfo>
                    <X509Data>
                        <X509Certificate>MIIFCDCCAvCgAw.........XdhRXasm+AZhWTGovAAqeLvDyjpAg=</X509Certificate>
                    </X509Data>
                </KeyInfo>
            </Signature>
        </GetTaxpayersRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.8.5.2 Response XML

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <SOAP-ENV:Body>
      <ns2:GetTaxpayersResponse Id="Response" Version="1" xmlns:ns2="https://Einvoice.tatime.gov.al/EinvoiceService/schema"
xmlns:ns3="http://www.w3.org/2000/09/xmldsig#">
         <ns2:Header RequestUUID="e7f25589-3882-4270-ae0c-05d2da18fb73" SendDateTime="2020-09-14T15:02:57Z" UUID="3799b13d-421b-44a5-a874-
6c3c063e0df0"/>
         <Taxpayers>
             <Taxpayer Name="Company name 1" Tin="A12345678A" Address="Rua 1" Town="Tirana" Country="ALB">
             <Taxpayer Name="Computing company" Tin="B12345678B" Address="Rua 2" Town="Tirana" Country="ALB">
         </Taxpayers>
         <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>3Cep3Ap4EEZ/.........AhosJtELsXF0=</DigestValue>
               </Reference>
            </SignedInfo>
            <SignatureValue>VJBiOpoSjZTqJ3NXNw+Os.........7ieoyJP9hTnKA==</SignatureValue>
            <KeyInfo>
               <X509Data>
                  <X509Certificate>MIIFRzCCBC+gAwIBAgI.........4RjtKkBTwYahJEZb8DE3jc9BSYwpAaAE=</X509Certificate>
               </X509Data>
            </KeyInfo>
         </Signature>
      </ns2:GetTaxpayersResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.9 ERROR MESSAGES

In case of an error in the process of the request message, an error message is sent as a reply by CIS. Error messages share the same general format which is based on SOAP fault message version 1.1 and extended with the code XML element which represents error code.

3.9.1 XML FORMAT

Name	Occurrence [Min, Max]	Description
fault	[1, 1]	XML element representing error message.
faultCode	[1, 1]	XML element representing class of errors.
faultString	[1, 1]	XML element where the error explanation is written.
detail	[1, 1]	XML element that carries error messages. It can contain multiple child elements.
responseUUID	[1, 1]	XML element that specifies UUID of this error response.

		requestUUID	[0, 1]	XML element that specifies UUID of the request for which error occurred if available.
		code	[1, 1]	XML element that describes the error code. List of codes can be found in the chapter 3.9.3.

Table 68

3.9.1.1 Header

This is an XML root element representing the header of the error message.

3.9.1.2 Header UUID

This is an attribute that uniquely describes the message and gives it the unique identification.

Data type	string
Length	36 characters
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 69

3.9.1.3 Fault

This is an XML element that will appear only if error happens.

3.9.1.4 FaultCode

This is an XML element that represents origin of error that occurred.

Data type	string
Constraint	Enumeration, described in the table below.
Example	Client

Table 70

Enumeration values for the method of payment are listed in table below.

Value	Description
Client	Received message was incorrectly formed or contained incorrect information.
Server	There was a problem with the server, so the message could not proceed.

Table 71

3.9.1.5 FaultString

This is an XML element that contains textual explanation for error that occurred.

Data type	string
Length	Undefined
Example	Validation failed with digest wrong.

Table 72

3.9.1.6 Detail

This is an XML element that carries numeric error code.

3.9.1.7 Code

This is a Detail’s child element, that describes the numeric error code. Numeric error codes are listed in the chapter 3.9.2.

Data type	String
Length	Undefined
Example	E_XML_FORMAT

Table 73

3.9.1.8 ResponseUUID

This is a Detail’s child element, that specifies UUID of this error message.

Data type	string
Length	36 characters
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 74

3.9.1.9 RequestUUID

This is a Detail’s child element, that specifies UUID of the request message that generated an error if available.

Data type	string
Length	36 characters
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 75

3.9.2 ERROR CODES

Following table lists the error codes that an einvoice service can return to the einvoice issuer. It's not an exhaustive list.

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.
12	Client	Received XML message failed schematron 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 keyInfo 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 NAIS.
36	Client	Certificate has expired.
37	Client	Compare the NIPT in XML with the NIPT in the certificate
38	Client	Certificate status revoked
39	Client	Certificate status unknown
40	Client	Private key used to sign request data message is not the same as private key used to sign einvoice data.
50	Client	Einvoice not supported
51	Client	Einvoice extraction failed
52	Client	Taxpayer doesn't exist in the Registry of taxpayers.
53	Client	Einvoice is missing.

54	Client	TIN inside request certificate is not the same as TIN inside einvoice data.
55	Client	Taxpayer is not active in the Registry of taxpayers.
9xx	Client	Undefined exceptions triggered by invalid request message.
9xx	Server	Internal server exceptions.

Table 76

3.9.3 EXAMPLE XML

```xml
<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>78dde160-2b33-40e4-98fa-f6a2c34475a3</requestUUID>
            <responseUUID>9416dcca-499a-4724-933d-40d115ea4fc7</requestUUID>
         </detail>
      </SOAP-ENV:Fault>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

4. Security

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

4.1 DIGITAL CERTIFICATES

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

- Certificate for transport security with common name *.tatime.gov.al issued to the service.
  Used to secure transport between the service and an issuer.
  Issued by public CA different from NAIS CA.
- Certificate for message security issued to the service.
  Used to digitally sign response data messages.
  Issued by NAIS CA.
- Certificate for message security issued to an issuer.
  Used to digitally sign request data messages and einvoice data.
  Issued by NAIS CA.

A digital certificate for message security is issued by the competent authority for issuing digital certificates (CAs). In the case of the Republic of Albania it is NAIS. A detailed description of the submission of a request for issuing a digital certificate and obtaining a certificate is (or will have to be) defined in the instructions of NAIS and is not the subject of this documentation. Certificates issued by NAIS will have following certificate authorities:

- NAIS Root Certification Authority (Self signed certificate)
    - NAIS Certification Authority
        - NAIS Class 3 Certification Authority

4.2 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 public CA different from NAIS CA, and with common name *.tatime.gov.al.

4.3 MESSAGE SECURITY

To ensure unambiguous identification of the taxpayer and to provide unchanged content, each request data message and einvoice data 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.

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 4.3.1.

Einvoice data is created and digitally signed according to the “European standard EN 16931” specification, additional description can be found inside chapter 4.3.3.

4.3.1 REQUEST AND RESPONSE DATA MESSASGE SIGNING

Every request and response data message described in the chapter 3, 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#
- 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

4.3.2 RESPONSE DATA MESSAGE SIGNATURE VERIFICATION

To verify that a response is created by einvoice service. TCR should check that signature is valid. That certificate provided inside response message is issued by AKSHI, by verifying certificate chain described in the chapter 4.1. And that certificate is issued to einvoice service by checking that certificate contains CN field with value “GDT eFiskalizimi” inside DN field.

4.3.3 EINVOICE DATA SIGNING

Einovice encapsulated inside request message must be signed according to the “European standard EN 16931” specification. Private key used to sign enveloped einvoice must be the same as one used to sign request message.

5. ANNEX - Code examples

This chapter covers the code examples for specific actions.

5.1 REQUEST DATA MESSAGE SIGNATURE GENERATION CODE

5.1.1 JAVA EXAMPLE

This is the example for the generation of the signature in Java language. Variables are hardcoded as this is just an example.

```
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://Einvoice.tatime.gov.al/EinvoiceService/schema";
    public static final String XML_REQUEST_ELEMENT = "RegisterEinvoiceRequest";
    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 =
        "<RegisterEinvoiceRequest " +
        "       xmlns=\"https://Einvoice.tatime.gov.al/EinvoiceService/schema\" " +
        "       xmlns:ns2=\"http://www.w3.org/2000/09/xmldsig#\" " +
        "       Id=\"Request\" " +
        "       Version=\"3\">\r\n" +
        "    <Header>...</Header>\r\n" +
        "    <EinvoiceEnvelope>...</EinvoiceEnvelope>\r\n" +
        "</RegisterEinvoiceRequest>";
    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();
        }
    }
}
```

5.1.2 C# EXAMPLE

This is the example for the generation of the signature in .NET C# language. Variables are hardcoded as this is just an example.

```
using System;
using System.IO;
using System.Security.Cryptography;
using System.Security.Cryptography.X509Certificates;
using System.Security.Cryptography.Xml;
using System.Xml;

namespace EinvoiceSigningUtilityDotNet
{
    class SampleGenerateSignature
    {
        public const String XML_SCHEMA_NS = "https://Einvoice.tatime.gov.al/ EinvoiceService/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 =
            "<RegisterEinvoiceRequest " +
            "       xmlns=\"https://Einvoice.tatime.gov.al/EinvoiceService/schema\" " +
            "       xmlns:ns2=\"http://www.w3.org/2000/09/xmldsig#\" " +
            "       Id=\"Request\" " +
            "       Version=\"3\">\r\n" +
            "    <Header>...</Header>\r\n" +
            "    <EinvoiceEnvelope>...</EinvoiceEnvelope>\r\n" +
            "</RegisterEinvoiceRequest>";
        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);
                }
            }
        }
    }
}
```

5.2 RESPONSE DATA MESSAGE SIGNATURE VALIDATION CODE

5.2.1 JAVA EXAMPLE

This is the example for the validation of the signature in Java language. Variables are hardcoded as this is just an example.

```
import java.io.*;
import java.security.cert.*;
import java.util.*;

import javax.xml.crypto.*;
import javax.xml.crypto.dom.*;
import javax.xml.crypto.dsig.*;
import javax.xml.crypto.dsig.keyinfo.*;
import javax.xml.crypto.dsig.dom.*;
import javax.xml.parsers.*;

import org.w3c.dom.*;
import org.xml.sax.*;

public class SampleValidateSignature {

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

    public static final String XML_SCHEMA_NS = "https://Einvoice.tatime.gov.al/EinvoiceService/schema";
    public static final String XML_RESPONSE_ELEMENT = "RegisterEinvoiceResponse";
    public static final String XML_RESPONSE_ID = "Response";
    public static final String XML_SIG_METHOD = "http://www.w3.org/2001/04/xmldsig-more#rsa-sha256";
    public static final String XML_SIGNATURE_ELEMENT = "Signature";
    public static final String KEY_ALGORITHM = "RSA";

    private static final String RESPONSE_TO_VALIDATE =
        "<env:Envelope xmlns:env=\"http://schemas.xmlsoap.org/soap/envelope/\">\r\n" +
        "    <env:Header/>\r\n" +
        "    <env:Body>\r\n" +
        "        <RegisterEinvoiceResponse " +
        "               xmlns=\"https://Einvoice.tatime.gov.al/EinvoiceService/schema\" " +
        "               xmlns:ns0=\"http://www.w3.org/2000/09/xmldsig#\" " +
        "               Id=\"Response\" " +
        "               Version=\"3\">\r\n" +
        "            <Header>...</Header>\r\n" +
        "            <EIC>...</EIC>\r\n" +
        "            <Signature>..</Signature>\r\n" +
        "        </RegisterEinvoiceResponse>\r\n" +
        "    </env:Body>\r\n" +
        "</env:Envelope>";

    public static void main(String[] args) throws Exception {

        // Get DOC from String XML
        DocumentBuilderFactory docFactory = DocumentBuilderFactory.newInstance();
        docFactory.setNamespaceAware(true);
        DocumentBuilder docBuilder = docFactory.newDocumentBuilder();
        Document doc = docBuilder.parse(new InputSource(new StringReader(RESPONSE_TO_VALIDATE)));

        // Get signature node
        NodeList signatureNodeList = doc.getElementsByTagNameNS(XMLSignature.XMLNS, XML_SIGNATURE_ELEMENT);
        if (signatureNodeList.getLength() == 0) {
            throw new Exception(String.format("XML signature element %s not found.", XML_SIGNATURE_ELEMENT));
        }
        Node signatureNode = signatureNodeList.item(0);

        // Get signature element
        DOMStructure ds = new DOMStructure(signatureNode);
        XMLSignature signature;
        try {
            signature = xmlSigFactory.unmarshalXMLSignature(ds);
        } catch (MarshalException e) {
            throw new Exception("Signature extraction exception.", e);
        }

        // Get certificate from signature element
        X509Certificate cert;
        KeyInfo keyInfo = signature.getKeyInfo();
        if (keyInfo == null) {
            throw new Exception("KeyInfo element is missing.");
        } else {
            List<?> list = keyInfo.getContent();
            if (list.size() != 1) {
                throw new Exception("More than one KeyInfo element.");
            } else {
                Object info = list.get(0);
                if (!(info instanceof X509Data)) {
                    throw new Exception("KeyInfo element is not x509 certificate.");
                }
                SignatureMethod sm = signature.getSignedInfo().getSignatureMethod();
                X509Certificate certificate = (X509Certificate)((X509Data) info).getContent().get(0);
                if (!sm.getAlgorithm().equalsIgnoreCase(XML_SIG_METHOD) ||
                    !certificate.getPublicKey().getAlgorithm().equalsIgnoreCase(KEY_ALGORITHM)) {
                    throw new Exception("Valid certificate not found.");
                }
                cert = certificate;
            }
        }

        // Get response node
        NodeList responseNodeList = doc.getElementsByTagNameNS(XML_SCHEMA_NS, XML_RESPONSE_ELEMENT);
        if (responseNodeList.getLength() == 0) {
            throw new Exception(String.format("XML element %s not found.", XML_RESPONSE_ELEMENT));
        }
        Node responseNode = responseNodeList.item(0);

        DOMValidateContext dvc = new DOMValidateContext(cert.getPublicKey(), signatureNode);
        dvc.setIdAttributeNS((Element)responseNode, null, "Id");

        // Get signedInfo element
        SignedInfo signedInfo = signature.getSignedInfo();
        List<?> refList = signedInfo.getReferences();
        if (refList != null && refList.size() != 1) {
            throw new Exception(String.format("Only one signature reference allowed, provided %s.", refList.size()));
        }

        // Check methods
        Reference reference = (Reference)refList.get(0);
        if (reference.getURI() == null || !reference.getURI().equals("#" + XML_RESPONSE_ID)) {
            throw new Exception(String.format("Signature reference URI must be provided with value \"%s\".", XML_RESPONSE_ID));
        }

        if (!signedInfo.getSignatureMethod().getAlgorithm().equalsIgnoreCase(XML_SIG_METHOD)) {
            throw new Exception(String.format("Signature method should be %s, provided %s.", XML_SIG_METHOD,
signedInfo.getSignatureMethod().getAlgorithm()));
        }

        if (!signedInfo.getCanonicalizationMethod().getAlgorithm().equalsIgnoreCase(CanonicalizationMethod.EXCLUSIVE)) {
            throw new Exception(String.format("Canonicalization method should be %s, provided %s.", CanonicalizationMethod.EXCLUSIVE,
signedInfo.getCanonicalizationMethod().getAlgorithm()));
        }

        if(!reference.getDigestMethod().getAlgorithm().equalsIgnoreCase(DigestMethod.SHA256)) {
            throw new Exception(String.format("Digest method should be %s, provided %s.", DigestMethod.SHA256,
reference.getDigestMethod().getAlgorithm()));
        }

        // Validate signature value
        try {
            boolean isValid = signature.validate(dvc);
            if (!isValid) {
                boolean signatureValueStatus = signature.getSignatureValue().validate(dvc);
                if (!signatureValueStatus) {
                    throw new Exception("Validation failed with signature value wrong.");
                }
                boolean referenceStatus = reference.validate(dvc);
                if (!referenceStatus) {
                    throw new Exception("Validation failed with digest wrong.");
                }
            }
        } catch (XMLSignatureException e) {
            throw new Exception("Signature validation failed.", e);
        }

        // Do additional checks
        //   * Check that certificate is issued by AKSHI CA
        //   * Check that CN of the certificate contains value "GDT eFiskalizimi"
        // Conclusion
        System.out.println ("Signature is valid");
    }
}
```

6. ANNEX - UBL Invoice to Fiscalization Fields Mapping

Following table describes relations bettween UBL Inovice fields and fiscalization service fields. Some fields in the e-Invoice according to UBL schema are the same as certain fields in fisclaization message schema, some fields are needed just in the e-Invoice UBL schema and will not be sent to fiscalization, and some fields are needed just for fiscalization and are not an integral part of the e-Invoice UBL shema. Each field and correlation is described in the table bellow.

E-INVOICE			FISCALIZATION MESSAGE			REMARKS
ID UBL	FIELD NAME UBL	DESCRIPTION UBL	ID FS	FIELD NAME	DESCRIPTION	
BT-1	Invoice number	Unique invoice identification	InvoiceNumber	Invoice number	Ordinal invoice number	-
-	-	-	InvOrdNum	Invoice ordinal number	Ordinal number of the invoice	Just in the fiscalization message
BT-2	Invoice Issue Date	Date when invoice was issued.	IssueDateTime	Date and time of invoice issuance	Date of invoice issuance	In the field BT-2 just the date must be written, and the time must be written in field BT-22
BT-3	Invoice Type Code	The code that determines the functional type of the invoice	CorrectiveInv Type	Type of the corrective invoice.	Type of corrective invoice	In the field BT-3 should be entered the relevant code from the code list representing the invoice type. When the invoice is a corrective invoice, than the code representing the corrective invoice/debit note/credit note should be entered.
-	-	-	TypeofInv	Type of the invoice (cash, non-cash).	Type of invoice (IN CASE OF E-Invoice it will be on-cash)	Just in the fiscalization message
-	-	-	TypeOfSelfIss	Type of self-invoice	Entered only if invoice is self-issued.	Just in the fiscalization message. In the field BT-3 the code for „self-billing“ must be entered.
-	-	-	IsSimplifiedInv	Is simplified invoice	Invoice issued is a simplified invoice.	Just in the fiscalization message. For e-invoice this field has always the attribute „false“.
BT-5	Invoice Currency Code	The currency in which all the invoice amounts are listed except the total amount of VAT in the accounting currency.	Currency Code	Currency Code	Currency code in which the amount on the invoice is payed (i.e. invoice is issued)	-
BT-6	Currency Code of VAT Calculation	Currency used for VAT calculation and reporting purposes accepted or required in the Seller's country.	-	-	-	-
BT-7	VAT Date Effective	Date when VAT becomes applicable to the Seller and Buyer, to the extent that this date may be determined and different from the date of issue of the invoice in accordance with the VAT Directive.	-	-	-	-
BT-8	Code for VAT Effective Date	The code for the date when VAT becomes effective for the Seller and Buyer.	-	-	-	-
BT-9	Payment Due Date	Date when payment is due.	PayDeadline	Payment deadline	Last day for payment.	-
BT-10	Buyer Reference	Identifier assigned by the Buyer for internal routing purposes.	-	-	-	-

E-INVOICE			FISCALIZATION MESSAGE			REMARKS
ID UBL	FIELD NAME UBL	DESCRIPTION UBL	ID FS	FIELD NAME	DESCRIPTION	
BT-11	Project Reference	Identification of the project to which the invoice relates	-	-	-	-
BT-12	Contract Reference	Identification of the contract.	-	-	-	-
BT-13	Purchase Order Reference	Identifier of the specified Purchase Order, issued by the Buyer.	-	-	-	-
BT-14	Sales Order Reference	The Sales Order Reference Identifier issued by the Seller.	-	-	-	-
BT-15	Receipt Reference	The receipt reference identifier.	-	-	-	-
BT-16	Despatch Note Reference	The shipping note reference identifier.	-	-	-	-
BT-17	Request for a Bid or Partial Bid Reference	Identification of a request for a bid or identification of a partial bid to which the invoice relates.	-	-	-	-
BT-18	Invoiced Item Identifier	The Item identifier upon which the invoice given by the Seller is based	-	-	-	-
	Schema Identifier	Identifier of the identification scheme of the Invoiced Item identifier.	-	-	-	-
BT-19	Charge Center	The textual value that determines where the relevant data will be entered in the Buyer's financial accounts.	-	-	-	-
BT-20	Payment Terms	A textual description of payment terms that applies to the amount due for payment (including a description of possible penalties).	-	-	-	-
BG-1	**INVOICE NOTE**					
BT-21	Item Code (heading) of textual notes	Item (heading) Text Notes in BT-22.	-	-	-	-
BT-22	Invoice Note	A text note that provides unstructured information relevant to an invoice as a whole.	IssueDateTime	Time of invoice issuance	Time	In BT-22 just the time of invoice issuance is written, since the date is written in field BT-2
			UII	Unique Invoice Identifier	Unique identifier of each invoice	-
			OperatorCode	Operator´s code	Reference to the operator code, who is operating on TCR and issues invoices.	-
			BusinUnitCode	Business premise´s code	The unique code of the business premise	-
			IIC	Issuer's invoice code	The Issuers unique code	-

E-INVOICE			FISCALIZATION MESSAGE			REMARKS
ID UBL	FIELD NAME UBL	DESCRIPTION UBL	ID FS	FIELD NAME	DESCRIPTION	
			Currency ExRate	Exchange rate	Exchange rate applied to calculate the equivalent amount of foreign currency for the total amount expressed in ALL. Exchange rate express equivalent amount of ALL for 1 unit of foreign currency.	Just if the invoice currency in field BT-5 is not ALL
			Currency IsBuying	Exchange transaction	True if exchange transaction is buying of the currency. False if exchange transaction is selling of the currency.	Just for currency exchange offices
			BadDebtInv	Bed debt invoice	The invoice is declared as a „bad debt invoice “	In the field BT-22 should be written „Bed debt invoice “, and in BT-25 the IIC of the original invoice is entered and in BT-26 the issuance date of the original invoice.
-	-	-	SoftCode	Software code	The code of the software solution for fiscalization	Just in the fiscalization message.
-	-	-	IICSignature	Signed issuer's invoice code concatenated parameters	Signed issuer's invoice code concatenated parameters	Just in the fiscalization message.
BG-2	**PROCESS CONTROL**					
BT-23	Business Process Type	Identifies the context of the business process in which the transaction occurs, in order to allow the Buyer to process an invoice appropriately.	-	-	-	-
BT-24	Identifier Specifications	Identification of specifications that contain the total set of semantic content rules, cardinalities and business rules with which the information contained in the document instance is consistent.	-	-	-	-
BG-3	**REFERENCE TO A PREVIOUS INVOICE**					
BT-25	Reference to a previous invoice	Identification of an invoice previously sent by the Seller.	CorrectiveInv IICRef	IIC reference on the original invoice	IIC reference on the original invoice which is being corrected	For corrective invoices. In BT-3 must be the code for corrective invoice, debit or credit note
			BadDebtInv IICRef	IIC reference on the original invoice	IIC reference on the original invoice which is declared as „bad debt“	For „Bed debt invoices “. In BT-22 must be: Bed debt invoice
			SumInvIICRef IICRef	IIC of the invoice that is referenced in the summary invoice.	For type of invoice: summary invoice.	In BT-3 the code for „consolidated invoice“ must be entered
BT-26	Issue date of previous invoice	The date when the previous invoice was issued.	CorrectiveInv IssueDateTime	Date of the original invoice	For corrective invoices	For corrective invoices. In BT-3 must be the code for corrective invoice, debit or credit note
			BadDebtInv IssueDateTime	Date of the original invoice	For „Bed debt invoices “. In BT-22 must be: Bed debt invoice	For „Bed debt invoices “. In BT-22 must be: Bed debt invoice
			SumInvIICRef IssueDateTime	Date of the invoice that is referenced in the summary invoice.	For type of invoice: summary invoice	In BT-3 the code for „consolodated invoice“ must be entered

E-INVOICE			FISCALIZATION MESSAGE			REMARKS
ID UBL	FIELD NAME UBL	DESCRIPTION UBL	ID FS	FIELD NAME	DESCRIPTION	
BG-4	**SELLER**					
BT-27	Seller’s Name	Complete formal name by which the Seller is entered in the national register of legal persons, either as a taxpayer or otherwise trading as a person or persons.	Seller Name	Seller’s name.	Name of the seller	-
BT-28	Seller's Trade Name	The name by which the Seller is recognized, other than the formal Seller Name (also known as the Company Name).	-	-	-	-
-	-	-	IsIssuerInVAT	Issuer is in VAT register	Seller is subject to VAT	Just in the fiscalization message.
-	-	-	Seller IDType	Seller’s identification number type.	Te type od the seller´s identification number	Just in the fiscalization message.
BT-29	Seller Identifier	Identification of Seller. For many systems, Seller's identifier is key information. Multiple vendor identifiers can be assigned or	Seller IDNum	Seller’s identification number.	The identification number of the seller.	-
	Schema identifier	Identifier of the identification scheme of the Seller identifier. If used, the identifier of the identification scheme will be selected from the list published by the ISO 6523 maintenance agency.	-	-	-	-
BT-30	Seller's Legal Registration Identifier	Officially issued identifier which serves to identify the Seller as a legal entity or a natural person. If no identification scheme is specified, it must be known to the Buyer and Seller.	Seller IDNum	Seller’s identification number.	The identification number of the seller.	-
	Schema identifier	Identifier of Identification Scheme for a Seller’s legal registration Identifier. If used, the identification scheme is selected from the list published by the ISO 6523 maintenance agency	-	-	-	-
BT-31	Seller VAT Identifier	Seller's VAT identifier. VAT number with country code prefix. A supplier registered as a taxpayer must include his VAT number, unless he or she uses a tax representative.	Seller IDNum	Seller’s identification number.	The identification number of the seller.	For Albanian taxpayer the VAT identifier is AL+NIPT and is written for taxpayers (sellers) subject to VAT, except if the invoice is issued for „out of scope VAT “
BT-32	Seller Tax Registry Identifier	Local Identification of the Seller (Defined by the Seller's Address) for taxation or reference, which allows the Seller to indicate their registered tax status. This information may affect how the Buyer settles the payment (such as Social Security fees). For example, in some countries, if the Seller is not registered as a taxpayer, the Buyer is required to retain the charge of taxes and pay it on behalf of the Seller.	Seller IDNum	Seller’s identification number.	The identification number of the seller.	-
BT-33	Additional legal information about the Seller	Additional legal information relevant to the Seller. Such as core capital.	-	-	-	-
BT-34	Seller's Electronic Address	Identifies the Seller's email address to which an account-level response can be provided	-	-	-	-
	Schema identifier	Identifier schema of the electronic address of the Seller. The schema identifier should be selected from the list maintained by the Connecting Europe Facility (CEF).	-	-	-	-
BG-5	**SELLER POSTAL ADDRESS**					
BT-35	Order of Seller Address, Line 1	Main address field. Usually street name and number or post office.	Seller Address	Seller’s address.	The address of the seller	-
BT-36	Order of Seller Address, Line 2	An additional address field that can be used to provide further details that	-	-	-	-
BT-162	Order of Seller Address, Line 3	An additional address field that can be used to provide further details that complement the main address field.	-	-	-	-
BT-37	City of Seller	The usual name of the place, town or settlement where the Seller's Address is located.	Seller Town	Seller’s town.	The town of the seller	-
BT-38	Seller's Postal Code	Identifier for a set of characters that can be addressed to the relevant postal service. Such as the postal number or postal code of the post office	-	-	-	-

E-INVOICE			FISCALIZATION MESSAGE			REMARKS
ID UBL	FIELD NAME UBL	DESCRIPTION UBL	ID FS	FIELD NAME	DESCRIPTION	
BT-39	Seller's County	County. Such as regions, counties, states, provinces, etc.	Seller Country	Seller's country.	The country of the seller	-
BT-40	Seller's Country Code	A code that identifies the country. If no tax representative is specified, this is the country where VAT is required. Lists of valid states are registered with the ISO 3166-1 maintenance agency, "Country Name and Subdivision Codes".	-	-	-	-
BG-6	SELLER CONTACT					
BT-41	Contact Point of Seller	A contact point for a legal person or a natural person. Such as the person's name, contact identification, department or office identification.	-	-	-	-
BT-42	Seller Contact Phone Number	Telephone Number of Contact Point	-	-	-	-
BT-43	Seller's contact E-mail Address	E-mail address for a contact point	-	-	-	-
BG-7	THE BUYER					
BT-44	Buyer's Name	The Buyer's full name	Buyer Name	Buyer's name.	The Buyer's full name	-
BT-45	Buyer's Trade Name	The name under which the Buyer is known, aside from the legally registered Buyer name, (also known as the Company Name). Can be used if it differs from the Buyer's legal name.	-	-	-	-
-	-	-	Buyer IDType	Buyer's identification number type.	The type of identification number of the buyer.	Just in fiscalization message
BT-46	Buyer Identifier	Identifier of the Buyer. If no Scheme is specified, it must be known to Buyer and Seller.	Buyer IDNum	Buyer's identification number.	The identification number of the buyer.	-
	Schema identifier	Identifier of the identification scheme for Customer Identification.If used, the identification scheme must be selected from the list of entries published by the ISO 6523 maintenance agency.	-	-	-	-
BT-47	Identifier for a legally registered Buyer	If no identification scheme is specified, the Buyer and the Seller must know it.	Buyer IDNum	Buyer's identification number.	The identification number of the buyer.	-
	Schema identifier	Identifier of the identification scheme for Customer Identification. If used, the identification scheme must be selected from the list of entries published by the ISO 6523 maintenance agency.	-	-	-	-
BT-48	Buyer Tax Identifier	Buyer's Tax Identifier (also known as Buyer's Tax Number). The tax number is prefixed by a country code according to ISO 3166-1.	Buyer IDNum	Buyer's identification number.	The identification number of the buyer.	-
BT-49	Buyer's Electronic Address	Identifies the Buyer's electronic address to which invoices are delivered.	-	-	-	-
	Schema identifier	Identifier of the Identification Scheme for a Buyer's Electronic Address. The schema identifier should be selected from the list which is maintained by the Connecting Europe Facility (CEF).	-	-	-	-
BG-8	BUYER POSTAL ADDRESS					
BT-50	Buyer Address, Line 1	Main address field. Usually street name and number or post office.	Buyer Address	Buyer's address.	Buyer's address.	-
BT-51	Buyer Address, Line 2	An additional address field that can be used to provide further details that complement the main address field.	-	-	-	-
BT-163	Buyer Address, Line 3	An additional address field that can be used to provide further details that complement the main address field.	-	-	-	-
BT-52	City of Buyer	The usual name of the place, city or settlement where the Buyer's address is located.	Buyer Town	Buyer's town.	Buyer's town.	-
BT-53	Buyer Postal Code	Identifier for a set of characters that can be addressed to the relevant postal service. Such as the postal number or postal code of the post office	-	-	-	-
BT-54	County of Buyer	County. Such as regions, counties, states, provinces, etc.	Buyer Country	Buyer's country.	Buyer's country.	-

E-INVOICE			FISCALIZATION MESSAGE			REMARKS
ID UBL	FIELD NAME UBL	DESCRIPTION UBL	ID FS	FIELD NAME	DESCRIPTION	
BT-55	Country code of Buyer	Lists of valid states are registered with the ISO 3166-1 maintenance agency, "Country Name and Subdivision Codes".	-	-	-	-
BG-9	BUYER CONTACT					
BT-56	Buyer contact point	A contact point for a legal or a natural person. Such as the person's name, contact identification, department or office identification.	-	-	-	-
BT-57	Buyer's Contact Phone Number	Contact point telephone number.	-	-	-	-
BT-58	Buyer's Contact Email Address	Contact point email address.	-	-	-	-
BG-10	PAYEE					
BT-59	Payee Name.	Payment Name Name. It is used when the payee differs from the Seller. However, the name of the payee can be the same as the Seller name.	-	-	-	-
BT-60	Payee Identifier	Payee Identifier. If no Scheme is specified, Buyer and Seller must know it.	-	-	-	-
	Schema identifier	Identifier of the Identification Scheme for the Payee Identifier. If used, the identification scheme is selected from the list which is published by the ISO 6523 maintenance agency.	-	-	-	-
BT-61	Identifier of legally registered Payee	Identifier issued by an official registrar that identifies the Payee as a legal or physical person. If no Scheme is specified, it must be known to Buyer and Seller.	-	-	-	-
	Schema identifier	Identifier of the identification scheme for identifying the legal registration of a payee. If used, the identification scheme is selected from the list published by the ISO 6523 maintenance agency.	-	-	-	-
BG-11	SELLER TAX REPRESENTATIVE					
BT-62	Name of Seller's Tax Representative	Full name of the Seller's tax representative.	-	-	-	-
BT-63	Tax Number of Seller's Tax Representative	Tax Identifier of the Seller's tax representative. Tax Number preceded by country code according to ISO 3166-1.	-	-	-	-
BG-12	ADDRESS OF SELLER'S TAX REPRESENTATIVE					
BT-64	Address of Seller's Tax Representative, Line 1	Main address field. Usually the name and street number or post office.	-	-	-	-
BT-65	Address of Seller's Tax Representative, Line 2	An additional address field that can be used to provide additional details that complement the main address field.	-	-	-	-
BT-164	Address of Seller's Tax Representative, Line 3	An additional address field that can be used to provide further details that complement the main address field.	-	-	-	-
BT-66	City of Seller's Tax Representative	The usual name for a place, city or settlement where the Address of the tax representative is located.	-	-	-	-
BT-67	Postal Code of Seller's Tax Representative	Identifier for a set of characters that can be addressed to the relevant postal service. Such as the postal number or postal code of the post office	-	-	-	-
BT-68	County of Seller's Tax Representative	County. Such as regions, counties, states, provinces, etc.	-	-	-	-
BT-69	Country Code of Seller's Tax Representative	A code that identifies the country..Lists of valid states are registered with the ISO 3166-1 maintenance agency, "Country Name and Subdivision Codes".	-	-	-	-
BG-13	DELIVERY INFORMATION					
BT-70	Shipping on Behalf of a Party	The name of the party to whom the goods and services are supplied. Used if delivery is to a customer different from the Buyer.	-	-	-	-
BT-71	Delivery Location Identifier	If no Scheme is specified, it must be known to the Buyer and Seller.	-	-	-	-
BT-72	Actual Delivery Date	Date of execution or completion of delivery of goods or services.	-	-	-	-
BG-14	INVOICE PERIOD					
BT-73	Invoice Period Start Date	The starting date of the billing period Initial delivery date of goods or services	SupplyDateOrPeriod Start	Start day of the supply.	-	-
BT-74	Invoice Period End Date	End date of the billing period. Date of delivery of goods or execution of services.	SupplyDateOrPeriod End	End day of the supply.	-	-
BG-15	DELIVERY ADDRESS					

E-INVOICE			FISCALIZATION MESSAGE			REMARKS
ID UBL	FIELD NAME UBL	DESCRIPTION UBL	ID FS	FIELD NAME	DESCRIPTION	
BT-75	Delivery Address, Line 1	Main address field. Usually the street name and number.	-	-	-	-
BT-76	Delivery Address, Line 2	An additional address field that can be used to provide further details that complement the main address field.	-	-	-	-
BT-165	Delivery Address, Line 3	An additional address field that can be used to provide further details that complement the main address field.	-	-	-	-
BT-77	Delivery City	Common name of the city, town or village where the Delivery Address is located	-	-	-	-
BT-78	Postal Code of Delivery	Identifier for a set of characters that can be addressed to the relevant postal service.	-	-	-	-
BT-79	Delivery County	County. Such as regions, counties, states, provinces, etc.	-	-	-	-
BT-80	Country Code of Delivery	A list of valid states is registered with the ISO 3166 1 maintenance agency, "Codes for the representation of names of countries and their subdivisions -- Part 1: Country codes".	-	-	-	-
BG-16	**PAYMENT INSTRUCTIONS**					
BT-81	Payment Means Code	The way, expressed as a code, in which payment is expected or has already been made.	-	-	-	-
BT-82	Text for Payment Means	The way, expressed as text, for which payment is expected or already made.	PayMethod Type	Type of the payment method.	Type of the payment method.	-
BT-83	Allocation Information	The text value used to establish a link between payment and the invoices issued by the Seller.	-	-	-	If the payment method is bank account, the UII of the e-invoice must be entered in BT-83
-	-	-	PayMethod Amt	Amount payed by payment method in the ALL	Amount payed by payment method in the ALL	Just in fiscalization message
BG-17	**CREDIT TRANSFER**					
BT-84	Payment Account Identifier	A unique billing account identifier, at payment service providers, to which payments must be executed. Like IBAN (in case of SEPA payment) or national account number.	-	-	-	-
BT-85	Payment Account Name	The payment account name, at payment service providers, to which payments must be executed.	-	-	-	-
BT-86	Payment Service Provider Identifier	Identifier for the payment service provider at which the invoice to be paid is located. For example, BIC or National Clearing Code (NCC) where necessary. No identification scheme is used.	-	-	-	-
BG-18	**PAYMENT CARD INFORMATION**					
BT-87	Primary account number	The primary account number (PAN) used for payment. In accordance with card security payment standards, an invoice must never contain a full card number. At present, the Security Standards Council of PCI has defined the following: The first 6 digits and the last four digits represent the maximum number of digits to be displayed.	-	-	-	-
BT-88	Card Holder's Name	The name of the payer.	-	-	-	-
BG-19	**DIRECT DEBIT**					
BT-89	Authorization Reference Identifier	Unique identifier assigned by the payee in reference to direct debit authorization. It is used to inform a Buyer in advance of SEPA direct debit.	-	-	-	-
BT-90	Bank Creditor Identifier	Unique bank identifier referring to the payee or Seller, which is assigned by the payee's bank or Seller. It is used to inform a Buyer in advance of SEPA direct debit.	-	-	-	-
BT-90	Bank Creditor Identifier	Unique bank identifier referring to the payee or Seller, which is assigned by the payee's bank or Seller. It is used to inform a Buyer in advance of SEPA direct debit.	-	-	-	-
BT-91	Debit Account Identifier	Account which is charged directly by debit.	-	-	-	-
BG-20	**DOCUMENT-LEVEL DISCOUNTS**					
BT-92	Amount of Document-level Discount	Amount of discount, without VAT.	-	-	-	-
BT-93	Basis of Document-level Discount	The basis which, together with the percentage of document-level discounts, is used to calculate the amount of a document-level discount.	-	-	-	-
BT-94	Percentage of Document-level Discount	The percentage which, together with the basis of a document-level discount, is used to calculate a document-level discount amount.	I.R	Rebate	Percentage of the rebate.	In the fiscalization message the discount is always written at the item level. If the discount in the fields BG-20 is entered, it always reduces the taxbase. If a discount is

E-INVOICE			FISCALIZATION MESSAGE			REMARKS
ID UBL	FIELD NAME UBL	DESCRIPTION UBL	ID FS	FIELD NAME	DESCRIPTION	
						given by the seller which does not reduces the tax base then in e-Invoice the amount of discount is written in the field BT-113 „Amount paid"
BT-95	VAT Category Code of Document-level Discount	Encoded VAT category identification which applies to a document-level discount. The following entries are applied UNTDID 5305 [6]: ● Standard rate ● Zero rated goods ● Exempt from VAT / IGIC / IPSI ● VAT Reverse charge / IGIC / IPSI ● VAT exempt for intra community supply of goods / IGIC / IPSI ● Free export item, tax not charged VAT / IGIC / IPSI because of export outside the EU ● Services outside scope of tax. (Non-taxable, Sales are not subject to VAT / IGIC / IPSI	-	-	-	-
BT-96	VAT Rate of Document-level Discount	The VAT rate is shown as the percentage which applies to a document-level discount.	-	-	-	-
BT-97	Reason for Document-level Discount	The reason for the document-level discount as expressed in text.	-	-	-	-
BT-98	Code for Reason for Document-level Discount	The reason for the document-level discount as expressed in code. Use UNTDID 5189 list entries [6]. The code which describes reason for the document-level discount, and the reason for the document-level discount must apply to the same document-level discount reason.	-	-	-	-
BG-21	**DOCUMENT-LEVEL CHARGES**					
BT-99	Document-Level Charge Amount	Charge amount without VAT.	-	-	-	In the fiscalization message the charges are written as a separate item, ori n the „fee" fiield for specific fees
BT-100	Base for Document-level Charge	A base that, together with the document-level charge percentage, can be used to calculate the amount of document-level charges.	-	-	-	-
BT-101	Document-level Charge Percentage	The percentage which, together with the basis of a document-level charge, is used to calculate a document-level charge amount.	-	-	-	-
BT-102	VAT Category Code of Document-level Charge	Encoded VAT category identification which applies to a document-level charge. The following entries are applied UNTDID 5305 [6]: ● Standard rate ● Zero rated goods ● Exempt from VAT / IGIC / IPSI ● VAT Reverse charge / IGIC / IPSI ● VAT exempt for intra community supply of goods / IGIC / IPSI ● Free export item, tax not charged VAT / IGIC / IPSI because of export outside the EU ● Services outside scope of tax. (Non-taxable, Sales are not subject to VAT / IGIC / IPSI	-	-	-	-
BT-103	VAT Rate of Document-level Charge	The VAT rate is shown as a percentage that applies to a document-level charge.	-	-	-	-
BT-104	Reason for Document-level Charge	The reason for a document-level charge as expressed in text.	-	-	-	-
BT-105	Code for Reason for Document-level Charge	The reason for a document-level charge as expressed in code.Use UNTDID 5189 list entries [6]. The code for a reason for document-level discount and the reason for the document-level discount must express the same reason for the discount.	-	-	-	-
BG-22	**TOTAL AMOUNTS**					
BT-106	Total of all Net Amounts for Invoice Items	Total of all net amounts for items on an invoice	-	-	-	-
BT-107	Total of Document-level Discounts	Total of all Document-level discounts on an invoice.Item-level discounts are included in the net amount of an itemized invoice and are summed into the total net amount of the invoice.	-	-	-	-
BT-108	Total of Document-level Charges	Total of all document-level Charges on an invoice.Item-level Charges are included in the net amount of an itemized invoice and are summed into the total net amount of the invoice.	-	-	-	-
BT-109	Total Invoice Amount Without VAT	Total amount of an invoice, not including VAT.The total amount of an invoice without VAT is the total net amount of the invoice items, minus the total of the document-level discounts and plus the total of document-level Charges.	TotPriceWoVAT	Total price of the invoice excluding VAT.	Total price of the invoice excluding VAT.	-

E-INVOICE			FISCALIZATION MESSAGE			REMARKS
ID UBL	FIELD NAME UBL	DESCRIPTION UBL	ID FS	FIELD NAME	DESCRIPTION	
BT-110	Total VAT invoice amount	Total VAT amount for account.. The total amount of VAT is calculated as the sum of all taxes of all VAT categories.	TotVATAmt	Total VAT amount of the invoice.	Total VAT amount of the invoice.	-
BT-111	Total Amount of VAT in the Accounting Currency	It is used when the VAT calculation currency (BT-6) differs from the invoice currency code (BT-5) in accordance with Article 230 of Directive 2006/112/EC on VAT. The amount of VAT in the accounting currency is not used to calculate the total amount of the invoice.	-	-	-	-
BT-112	Total Amount of Invoice with VAT	The total amount of an invoice with VAT included.-The total amount of VAT invoiced is the total amount of and invoice without VAT, plus the total VAT amount for the invoice.	TotPrice	Total price of all items including taxes and discounts.		
BT-113	Amount Paid	Total of prepaid amounts.This amount is deducted from the total amount of the VAT invoice for the calculation of the due payment amount.	-	-	-	-
BT-114	Amount of Rounding	Amount which must be added to total in order to round off the payment amount.	-	-	-	-
BT-115	Amount Due for Payment	Remaining payment amount. This amount is the total amount of the invoice with VAT deducted for the prepaid amount. The amount is zero in the case of a fully paid invoice. The amount may be negative, in which case the Seller owes that amount to the Buyer.	-	-	-	-
BG-23	**DISTRIBUTION OF VAT**					
BT-116	Base Amount of VAT Category	The total of all taxable amounts subject to a particular VAT category code and VAT category rate (if VAT rate applies)..The net total of all items, minus discounts, plus document-level Charges that are subject to specific VAT category codes and VAT rate categories (if VAT rate applies)	SameTax.PriceBefVAT	Price before VAT	For all items of the same VAT category	The amount from BT-116 is written in the fiscalizaion message field „SameTax.PriceBefVAT" if in BT-118 the code is selected for: • Standard rate • Zero rated goods • Exempt from VAT / IGIC / IPSI • VAT Reverse charge / IGIC / IPSI
			TaxFreeAmt	Amount of sale without VAT	Amount of sale without VAT	The amount from BT-116 is written in the fiscalizaion message field „TaxFreeAmt" if: - in BT-118 Category Code, the category choosen is Exempt from VAT" or „Services outside scope of tax. (Non-taxable, Sales are not subject to VAT / IGIC / IPSI / IGIC / IPSI", and - in field BT-121 the code is not: VATEX-EU-132, VATEX-EU-D, VATEX-EU-F, VATEX-EU-I, VATEX-EU-J, and - in BT-120 the reason for exemption is not: Exempted on the basis of Articles 53 and 54 of the VAT law
			MarkUpAmt	Amount related to special procedure for margin sheme	Amount related to special procedure for margin sheme	The amount from BT-116 is written in the fiscalizaion message field „MarkUpAmt" if: - in the VAT Category Code (BT-118) the category choosen is: „Exempt from VAT", and - in field BT-121 the code is: VATEX-EU-D, VATEX-EU-F, VATEX-EU-I, VATEX-EU-J
			GoodsExAmt	Export of goods	Export of goods	The amount from BT-116 is written in the fiscalizaion message field „GoodsExAmt" if in the VAT Category Code (BT-118) the category choosen is „Free export item, tax not charged VAT / IGIC / IPSI because of export outside the EU"
BT-117	Amount of VAT Category	Total VAT amount for a particular VAT category.. Calculated by multiplying the taxable amount of VAT with the VAT rate category rate for the relevant VAT category.	SameTax.VATAmt	VAT amount	VAT amount	-
BT-118	VAT Category Code	Coded identification for VAT category.. The following entries are applied UNTDID 5305 [6]: • Standard rate • Zero rated goods • Exempt from VAT / IGIC / IPSI • VAT Reverse charge / IGIC / IPSI • VAT exempt for intra community supply of goods / IGIC / IPSI • Free export item, tax not charged VAT / IGIC / IPSI because of export outside the EU • Services outside scope of tax. (Non-taxable, Sales are not subject to VAT / IGIC / IPSI	IsReverseCharge	„Reverse Charge"	If true, the buyer is obliged to pay the VAT.	If in the BT-118 the „VAT Reverse Charge" is selected" then the field „IsReverseCharge" in the fiscalization mesage must be „true"
BT-119	VAT Rate Category	The VAT rate, represented by the percentage of the relevant VAT category.. The VAT category codes and VAT rate categories must be consistent.	SameTax.VATRate	VAT rate	For all items of the same VAT category	-
BT-120	Text on the Reason for VAT Exemption	A textual statement of reasons why the amount is exempt from VAT or why VAT has not been charged . Articles 226, paragraphs 11 to 15 of Directive 2006/112/EC	-	-	-	-
BT-121	Code for the Reasons for VAT Exemption	Encoded statement of reasons why the amount is exempt from VAT. List of Codes issued and maintained by the Connecting Europe Facility (CEF)	-	-	-	-

E-INVOICE			FISCALIZATION MESSAGE			REMARKS
ID UBL	FIELD NAME UBL	DESCRIPTION UBL	ID FS	FIELD NAME	DESCRIPTION	
BG-24	**ADDITIONAL SUPPORTING DOCUMENTS**					
BT-122	Reference to Supporting Document	Identifier of the supporting document.	-	-	-	-
BT-123	Description of Supporting Document	A text description of the supporting document.. Such as: evidence of time tracking, usage report, etc.	-	-	-	-
BT-124	Location of External Documents	URL (Uniform Resource Locator) identifying where external documents are located. Resource location mode, including its primary access mechanism, e.g., http://or ftp//. The external location of the document must be used if the Buyer requires additional supporting information about an invoice. External documents are not part of the invoice. There are risks in accessing external documents.	-	-	-	-
BT-125	Attached Document	An attached document embedded as a binary object or sent along with an invoice.. An attached document is used when the documentation has to be saved together with the invoice for future reference or for auditing purposes.	-	-	-	-
	Attached Mime Type Document	Mime types of enclosed document.. Allowed mime types: ● application/pdf ● image/png ● image/jpeg ● text/csv ● application/vnd.openxmlformats-officedocument.spreadsheetml.sheet application/vnd.oasis.opendocument.spreadsheet	-	-	-	-
	Attached Document File Name	File name of the attached document	-	-	-	-
BG-25	**INVOICE ITEM**					
BT-126	Invoice Item Identifier	A unique identifier for a single invoice item.	Item C (Code)	Code of the item from the barcode or similar representation	Code of the item from the barcode or similar representation	-
BT-127	Item Notes	A text note that gives unstructured information relevant to an invoice item.	-	-	-	-
BT-128	Specific Invoice Item Identifier	An identifier for a specific object on which an invoice item is based, provided by the Seller. It can be a subscription number, a phone number, a unit of measurement, etc., as required.	-	-	-	-
	Schema identifier	An identifier of the identification scheme for a specific invoice item's identifier. If the recipient from the identifier is not clear about which schema is used, a conditional schema identifier must be used which must be selected from the UNTDID 1153 code list [6].	-	-	-	-
BT-129	Invoiced Quantity	The quantity of items (goods or services) that are charged as invoice items.	Item Q (Quantity)	Amount or number (quantity) of items.	Amount or number (quantity) of items.	-
BT-130	Code for Unit of Measurement for Invoiced Quantities	Unit of measure relating to the invoiced quantity. The unit of measure must be selected from UN/ECE Recommendation No. 20 "Codes for Units of Measure Used in International Trade" [7] and UN/ECE Recommendation No. 21 "Codes for passengers, types of cargo, packages and packaging materials (with complementary codes for package names)" using the method described in Rec No.20 Intro 2.a). Please note that in most cases it is not necessary for Buyers and Sellers to implement their listings in their software in their entirety. Sellers should only support those units of measures needed for their goods and services. Customers only need to check that the units of measure used in the account are equal to the units of measures used in other documents (such as a contract, catalog, order and delivery note).	Item U (Unit of measure)	Measuring unit	The item's unit of measure (piece, weight measure, length measure, etc.)	-
BT-131	Net Amount of Invoice Items	The total amounts for invoice items. The amount is "net" without VAT, Including item-level discounts and Charges, as well as other relevant taxes.	I.PB	Item price before VAT	Quantity * unit price (BT-129*BT-146)	-
BT-132	Referenced Item Purchase Order reference	Identifier for a referenced item within a purchase order issued by the Buyer. The purchase order identifier is referred to at the document level.	-	-	-	-
BT-133	Buyer Accounting Reference on Item (Cost Center)	The textual value that determines where the relevant data will be posted to a Buyer's financial accounts. If necessary,	-	-	-	-

E-INVOICE			FISCALIZATION MESSAGE			REMARKS
ID UBL	FIELD NAME UBL	DESCRIPTION UBL	ID FS	FIELD NAME	DESCRIPTION	
		the Buyer will provide this Reference to the Seller before the invoice is issued.				
BG-26	**INVOICING PERIOD FOR INVOICE ITEMS**					
BT-134	Invoice Item Invoicing Period Start Date	The date when the invoicing period for an invoice item starts. Date which refers to the first day of the period.	-	-	-	-
BT-135	Invoice Item Invoicing Period End Date	The date when the invoicing period for an invoice item ends. Date which refers to the last day of the period.	-	-	-	-
BG-27	**DISCOUNTS ON INVOICE ITEMS**					
BT-136	Amount of Invoice Item Discount	Amount of discount without VAT.	-	-	-	-
BT-137	Base for Invoice Item Discount	The basic amount that, together with the discount percentage of the invoice item, can be used to calculate the discount amount of the invoice item.	-	-	-	-
BT-138	Percentage of Invoice Item Discount	Percentage that, with the base for an invoice item discount, can be used to calculate the discount amount of the invoice item	I.R	Rebate	Percentage of the rebate.	If the discount in the field BT-138 is entered, it always reduces the taxbase. If a discount is given by the seller which does not reduces the tax base, then in e-Invoice the amount of discount is written in the field BT-113 „Amount paid “
BT-139	Reason for Discount on Invoice Item	The reason for a discount on invoice items as expressed in text.	-	-	-	-
BT-140	Code for Reason for Discount on Invoice Item	The reason for discount on an invoice item expressed as a code.Use UNTDID 5189 code lists [6]. The code for reason for an item-level discount and the reason for item-level discount must refer to the same reason for the discount.	-	-	-	-
BG-28	**INVOICE ITEM CHARGE**					
BT-141	Amount of Invoice Item Charge	Amount charged without VAT.	-	-	-	-
BT-142	Base Charge for Invoice Item	The basic amount that, together with the Charge percentage of an invoice item, can be used to calculate the Charge of the invoice item.	-	-	-	-
BT-143	Charge Percentage of Invoice Item	Percentage that, together with the charge percentage of an invoice item, can be used to calculate the charge for the invoice item.	-	-	-	-
BT-144	Reason for Invoice Item Charge	The reason for the Charge of an invoice item as expressed in text.	-	-	-	-
BT-145	Code for Reason for Invoice Item Charge	The reason for an invoice item's Charge expressed as a code.Use the list of codes UNTDID 7161 [6]. The code for reason for an item-level Charge and the reason for item-level Charge must refer to the same reason for the Charge.	-	-	-	-
BG-29	**PRICE DETAILS**					
BT-147	Price Reduction (Discount Price)	Total discount deducted from the gross price of the item in order to calculate the net price of the item. Applies only if the reduction is provided per unit and is not included in the gross price of the item.	-	-	-	In the fiscalization message it is written in percentage at the level of the item
BT-146	Net Item Price	Item price without VAT after deduction of the discount price on an item. The net price of the item must be equal to the gross price of the item, reduced by the value of the discount on the item.	I.UPB	Unit price without VAT	Per each item, quantity=1	-
BT-149	Item Unit Price	Number of item units to which the price refers.	-	-	-	The quantity to which this net item price from the field BT-146 applies
BT-150	Code for Measuring a Quantity of Items as a Unit	Unit of measurement pertaining to the price of a combined quantity of items. The unit of measure for the price of a combined quantity of items must be the same as the measurement unit of the invoiced quantities.	I.U	Unit of measure	Unit of measure	-
-	-	-	I.PA	Price with VAT	I.PA=I.PB+I.VA	Just in the fiscalization message
-	-	-	I.UPA	Unit price with VAT	I.UPA=I.PA/Q	Just in the fiscalization message
BG-30	**VAT INFORMATION FOR INVOICE ITEMS**					
BT-151	VAT Category Code for Invoiced Items	The VAT category code for invoiced items. The following entries are applied UNTDID 5305 [6]: • Standard rate	I.EX	Exempt from VAT	Exempt from VAT	If in the field BT-151 is selected „Exempted from VAT" then at the item level in fiscalization message the field I.EX is

E-INVOICE			FISCALIZATION MESSAGE			REMARKS
ID UBL	FIELD NAME UBL	DESCRIPTION UBL	ID FS	FIELD NAME	DESCRIPTION	
		• Zero rated goods • Exempt from VAT / IGIC / IPSI • VAT Reverse charge / IGIC / IPSI • VAT exempt for intra community supply of goods / IGIC / IPSI • Free export item, tax not charged VAT / IGIC / IPSI because of export outside the EU Services outside scope of tax. (Non-taxable, Sales are not subject to VAT / IGIC / IPSI)				entered with the appropriate value (Type 1, Type 2
BT-152	VAT Rate for Invoiced Items	The VAT rate shown as a percentage of the invoice item.	I.VR	VAT rate	VAT rate applied at item level	-
-	-	-	I.VA	VAT amount per invoice item	I.VA=I.PB*I.VR	Just in the fiscalization message
BG-31	**ITEM INFORMATION**					
BT-153	Product Name	Product name.	Items N (Name)	Name of the item (goods or services).	Name of the item (goods or services).	-
BT-154	Item Description	Item description. The item description provides a more detailed description of the item and its features than the name of the item.	-	-	-	-
BT-155	Item Identifier of Seller	The identifier assigned to the item by the Seller.	Item C (Code)	Code of the item from the barcode or similar representation	Code of the item from the barcode or similar representation	-
BT-156	Item Identifier of Buyer	The identifier assigned to the item by the Buyer.	-	-	-	-
BT-157	Standard Item Identifier	An item identifier based on a registered scheme.	-	-	-	-
	Schema Identifier	Identifier of the identification scheme for a standard item identifier. The identification scheme can be found on the list published by the ISO/IEC 6523 maintenance agency.	-	-	-	-
BT-158	Item Classification Identifier	Code for classifying articles according to their type or nature. Classification codes are used to allow the grouping of similar items for various purposes, such as public procurement (CPV), e-Commerce (UNSPSC) etc.	-	-	-	-
	Schema Identifier	Identifier of the identification scheme for item classification identifiers. The identification scheme must be selected from UNTDID 7143 [6]	-	-	-	-
	Schema Version Identifier	Version of the identification scheme.	-	-	-	-
BT-159	Country of Origin of Item	The code identifying the country from which an item originated. A list of valid states is registered with the ISO 3166 1 "Codes for the representation of names of countries and their subdivisions".	-	-	-	-
BG-32	**ITEM ATTRIBUTES**					
BT-160	Item Attribute Name	Name of an attribute or property of an item.Such as "Color."	-	-	-	-
BT-161	Item Attribute Value	Value of attributes or properties of an article. Such as "Red".	-	-	-	-
-	-	-	VS.VD.D	Expiration date of the voucher.	Expiration date of the voucher.	Just in fiscalization message.
			VS.VD.N	Nominal voucher value.	Nominal voucher value.	Just in fiscalization message.
-	-	-	VS.VD.T	Voucher type	Voucher type	Just in fiscalization message.
-	-	-	VS.VN.V.Num	Voucher serial number	Voucher serial number	Just in fiscalization message.
-	-	-	Fees.Type	Type of the fee	Type of the fee	Just in fiscalization message.
-	-	-	Fees.Amt	Amount of the fee	Amount of the fee	Just in fiscalization message. In e-invoice the amount is written as a separate item or as „Charge “

7. ANNEX - WSDL version 1

```xml
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
    name="EinvoiceService"
    targetNamespace="https://Einvoice.tatime.gov.al/EinvoiceService"
    xmlns:al="https://Einvoice.tatime.gov.al/EinvoiceService"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:als="https://Einvoice.tatime.gov.al/EinvoiceService/schema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

    <wsdl:types>
        <xsd:schema>
            <xsd:import namespace="https://Einvoice.tatime.gov.al/EinvoiceService/schema" schemaLocation="alimc-einvoice-service.xsd"/>
        </xsd:schema>
    </wsdl:types>

    <wsdl:message name="RegisterEinvoiceRequest">
        <wsdl:documentation>Element representing register sale message.</wsdl:documentation>
        <wsdl:part element="als:RegisterEinvoiceRequest" name="request" />
    </wsdl:message>

    <wsdl:message name="RegisterEinvoiceResponse">
        <wsdl:documentation>Element representing register invoice response message.</wsdl:documentation>
        <wsdl:part element="als:RegisterEinvoiceResponse" name="response" />
    </wsdl:message>

    <wsdl:message name="EinvoiceChangeStatusRequest">
        <wsdl:documentation>Element representing Einvoice change status request message.</wsdl:documentation>
        <wsdl:part element="als:EinvoiceChangeStatusRequest" name="request" />
    </wsdl:message>

    <wsdl:message name="EinvoiceChangeStatusResponse">
        <wsdl:documentation>Element representing Einvoice change status response message.</wsdl:documentation>
        <wsdl:part element="als:EinvoiceChangeStatusResponse" name="response" />
    </wsdl:message>

    <wsdl:message name="GetTaxpayersRequest">
        <wsdl:documentation>Element representing get Taxpayers request message.</wsdl:documentation>
        <wsdl:part element="als:GetTaxpayersRequest" name="request" />
    </wsdl:message>

    <wsdl:message name="GetTaxpayersResponse">
        <wsdl:documentation>Element representing get Taxpayers response message.</wsdl:documentation>
        <wsdl:part element="als:GetTaxpayersResponse" name="response" />
    </wsdl:message>

    <wsdl:message name="GetEinvoicesRequest">
        <wsdl:documentation>Element representing get Einvocies request message.</wsdl:documentation>
        <wsdl:part element="als:GetEinvoicesRequest" name="request" />
    </wsdl:message>

    <wsdl:message name="GetEinvoicesResponse">
        <wsdl:documentation>Element representing get Einovices response message.</wsdl:documentation>
        <wsdl:part element="als:GetEinvoicesResponse" name="response" />
    </wsdl:message>

    <wsdl:portType name="EinvoiceServicePortType">
        <wsdl:operation name="registerEinvoice">
            <wsdl:input message="al:RegisterEinvoiceRequest"/>
            <wsdl:output message="al:RegisterEinvoiceResponse"/>
        </wsdl:operation>
        <wsdl:operation name="einvoiceChangeStatus">
            <wsdl:input message="al:EinvoiceChangeStatusRequest"/>
            <wsdl:output message="al:EinvoiceChangeStatusResponse"/>
        </wsdl:operation>
        <wsdl:operation name="getTaxpayers">
            <wsdl:input message="al:GetTaxpayersRequest"/>
            <wsdl:output message="al:GetTaxpayersResponse"/>
        </wsdl:operation>
        <wsdl:operation name="getEinvoices">
            <wsdl:input message="al:GetEinvoicesRequest"/>
            <wsdl:output message="al:GetEinvoicesResponse"/>
        </wsdl:operation>
    </wsdl:portType>

    <wsdl:binding name="EinvoiceServiceSoap" type="al:EinvoiceServicePortType">
        <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="registerEinvoice">
            <soap:operation soapAction="https://Einvoice.tatime.gov.al/EinvoiceService/RegisterEinvoice"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="einvoiceChangeStatus">
            <soap:operation soapAction="https://Einvoice.tatime.gov.al/EinvoiceService/EinvoiceChangeStatus"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
```

```
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="getTaxpayers">
            <soap:operation soapAction="https://Einvoice.tatime.gov.al/EinvoiceService/GetTaxpayers"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="getEinvoices">
            <soap:operation soapAction="https://Einvoice.tatime.gov.al/EinvoiceService/GetEinvoices"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>

    <wsdl:service name="EinvoiceService">
        <wsdl:port name="EinvoiceServicePort" binding="al:EinvoiceServiceSoap">
            <soap:address location="https://Einvoice.tatime.gov.al/EinvoiceService-v1"/>
        </wsdl:port>
    </wsdl:service>

</wsdl:definitions>
```

8. ANNEX - XSD version 1

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
    targetNamespace="https://Einvoice.tatime.gov.al/EinvoiceService/schema"
    xmlns:al="https://Einvoice.tatime.gov.al/EinvoiceService/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="RegisterEinvoiceRequest">
        <annotation>
            <documentation>Root XML element representing register sale message.</documentation>
        </annotation>
        <complexType>
            <all minOccurs="1" maxOccurs="1">
                <element name="Header" type="al:RegisterEinvoiceRequestHeaderType" minOccurs="1" maxOccurs="1">
                    <annotation>
                        <documentation>Element representing the header of the message.</documentation>
                    </annotation>
                </element>
                  <element name="EinvoiceEnvelope" type="al:EinvoiceEnvelopeType" minOccurs="1" maxOccurs="1">
                    <annotation>
                        <documentation>XML element containing actual einvoice XML in UBL or CII format.</documentation>
                    </annotation>
                  </element>
                <element ref="ds:Signature" minOccurs="1" maxOccurs="1">
                    <annotation>
                        <documentation>XML element representing a signature.</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="al:IntSType" use="required" fixed="1">
                <annotation>
                    <documentation>Attribute used to specify compliance with XSD schema.</documentation>
                </annotation>
            </attribute>
        </complexType>
    </element>
    <element name="RegisterEinvoiceResponse">
        <annotation>
            <documentation>Root XML element representing register invoice response message.</documentation>
        </annotation>
        <complexType>
            <all minOccurs="1" maxOccurs="1">
                <element name="Header" type="al:RegisterEinvoiceResponseHeaderType" minOccurs="1" maxOccurs="1">
                    <annotation>
                        <documentation>Element representing the header of the message.</documentation>
                    </annotation>
                </element>
                <element name="EIC" type="al:UUIDSType" minOccurs="1" maxOccurs="1">
                    <annotation>
                        <documentation>CIS generated verification code that can be used to uniquely identify registered 
einvoice.</documentation>
                    </annotation>
                </element>
                <element name="Warnings" type="al:WarningsType" minOccurs="0" maxOccurs="1" >
                    <annotation>
                        <documentation>XML element representing list of warnings.</documentation>
                    </annotation>
                </element>
                <element ref="ds:Signature" minOccurs="1" maxOccurs="1">
                    <annotation>
                        <documentation>XML element representing a signature.</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="al:IntSType" use="required" fixed="1">
                <annotation>
                    <documentation>Attribute used to specify compliance with XSD schema.</documentation>
                </annotation>
            </attribute>
        </complexType>
    </element>
    <element name="EinvoiceChangeStatusRequest">
        <annotation>
            <documentation>Root XML element representing Einvoice change status request message.</documentation>
        </annotation>
        <complexType>
            <all minOccurs="1" maxOccurs="1">
                <element name="Header" type="al:EinvoiceChangeStatusRequestHeaderType" minOccurs="1" maxOccurs="1">
```

```
                        <annotation>
                            <documentation>Element representing the header of the message.</documentation>
                        </annotation>
                    </element>
                    <element name="EICs" type="al:EICsType" minOccurs="1" maxOccurs="1">
                        <annotation>
                            <documentation>XML element representing list of Einvoice ids.</documentation>
                        </annotation>
                    </element>
                    <element name="EinStatus" type="al:EinvoiceStatusType" minOccurs="1" maxOccurs="1">
                        <annotation>
                            <documentation>Element that represents Einvoice status.</documentation>
                        </annotation>
                    </element>
                    <element ref="ds:Signature" minOccurs="1" maxOccurs="1">
                        <annotation>
                            <documentation>XML element representing a signature.</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="al:IntSType" use="required" fixed="1">
                    <annotation>
                        <documentation>Attribute used to specify compliance with XSD schema.</documentation>
                    </annotation>
                </attribute>
            </complexType>
        </element>
        <element name="EinvoiceChangeStatusResponse">
            <annotation>
                <documentation>Root XML element representing Einvoice change status response message.</documentation>
            </annotation>
            <complexType>
                <all minOccurs="1" maxOccurs="1">
                    <element name="Header" type="al:EinvoiceChangeStatusResponseHeaderType" minOccurs="1" maxOccurs="1">
                        <annotation>
                            <documentation>Element representing the header of the message.</documentation>
                        </annotation>
                    </element>
                    <element name="ResponseCode" type="al:ResponseCodeType" minOccurs="1" maxOccurs="1">
                        <annotation>
                            <documentation>Response code.</documentation>
                        </annotation>
                    </element>
                    <element ref="ds:Signature" minOccurs="1" maxOccurs="1">
                        <annotation>
                            <documentation>XML element representing a signature.</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="al:IntSType" use="required" fixed="1">
                    <annotation>
                        <documentation>Attribute used to specify compliance with XSD schema.</documentation>
                    </annotation>
                </attribute>
            </complexType>
        </element>
        <element name="GetTaxpayersRequest">
            <annotation>
                <documentation>Root XML element representing get Taxpayers request message.</documentation>
            </annotation>
            <complexType>
                <all minOccurs="1" maxOccurs="1">
                    <element name="Header" type="al:GetTaxpayersRequestHeaderType" minOccurs="1" maxOccurs="1">
                        <annotation>
                            <documentation>Element representing the header of the message.</documentation>
                        </annotation>
                    </element>
                      <element name="Filter" type="al:TaxpayerFilterType" minOccurs="1" maxOccurs="1">
                        <annotation>
                            <documentation>XML element containing filters.</documentation>
                        </annotation>
                      </element>
                    <element ref="ds:Signature" minOccurs="1" maxOccurs="1">
                        <annotation>
                            <documentation>XML element representing a signature.</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="al:IntSType" use="required" fixed="1">
                    <annotation>
                        <documentation>Attribute used to specify compliance with XSD schema.</documentation>
                    </annotation>
                </attribute>
            </complexType>
        </element>
        <element name="GetTaxpayersResponse">
            <annotation>
                <documentation>Root XML element representing get Taxpayers response message.</documentation>
            </annotation>
            <complexType>
                <all minOccurs="1" maxOccurs="1">
                    <element name="Header" type="al:GetTaxpayersResponseHeaderType" minOccurs="1" maxOccurs="1">
                        <annotation>
                            <documentation>Element representing the header of the message.</documentation>
                        </annotation>
                    </element>
                    <element name="Taxpayers" type="al:TaxpayersType" minOccurs="1" maxOccurs="1">
                        <annotation>
                            <documentation>XML element representing list of taxpayers.</documentation>
                        </annotation>
                    </element>
                    <element ref="ds:Signature" minOccurs="1" maxOccurs="1">
                        <annotation>
                            <documentation>XML element representing a signature.</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="al:IntSType" use="required" fixed="1">
                    <annotation>
                        <documentation>Attribute used to specify compliance with XSD schema.</documentation>
                    </annotation>
                </attribute>
            </complexType>
        </element>
        <element name="GetEinvoicesRequest">
            <annotation>
                <documentation>Root XML element representing get Einvoices request message.</documentation>
            </annotation>
            <complexType>
                <all minOccurs="1" maxOccurs="1">
                    <element name="Header" type="al:GetEinvoicesRequestHeaderType" minOccurs="1" maxOccurs="1">
                        <annotation>
                            <documentation>Element representing the header of the message.</documentation>
                        </annotation>
                    </element>
                    <element name="EIC" type="al:UUIDSType" minOccurs="0" maxOccurs="1">
                        <annotation>
                            <documentation>Einvoice EIC for find One Einvoice, only find one returns pdf.</documentation>
                        </annotation>
                    </element>
                    <element name="PartyType" type="al:PartyTypeSType" minOccurs="0" maxOccurs="1">
                        <annotation>
                            <documentation>Einvoice party type (Buyer or Seller)</documentation>
                        </annotation>
                    </element>
                    <element name="RecDateTimeFrom" type="al:UTCSType" minOccurs="0" maxOccurs="1">
                        <annotation>
                            <documentation>Receive date time from</documentation>
                        </annotation>
                    </element>
                    <element name="RecDateTimeTo" type="al:UTCSType" minOccurs="0" maxOccurs="1">
                        <annotation>
                            <documentation>Receive date time to</documentation>
                        </annotation>
                    </element>
                    <element ref="ds:Signature" minOccurs="1" maxOccurs="1">
                        <annotation>
                            <documentation>XML element representing a signature.</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="al:IntSType" use="required" fixed="1">
                    <annotation>
                        <documentation>Attribute used to specify compliance with XSD schema.</documentation>
                    </annotation>
                </attribute>
            </complexType>
        </element>
        <element name="GetEinvoicesResponse">
            <annotation>
                <documentation>Root XML element representing get Einvoices response message.</documentation>
            </annotation>
            <complexType>
                <all minOccurs="1" maxOccurs="1">
                    <element name="Header" type="al:GetEinvoicesResponseHeaderType" minOccurs="1" maxOccurs="1">
                        <annotation>
                            <documentation>Element representing the header of the message.</documentation>
                        </annotation>
                    </element>
                    <element name="Einvoices" type="al:EinvoicesType" minOccurs="1" maxOccurs="1">
                        <annotation>
```

```
                        <documentation>XML element representing list of Einvoices.</documentation>
                    </annotation>
                </element>
                <element ref="ds:Signature" minOccurs="1" maxOccurs="1">
                    <annotation>
                        <documentation>XML element representing a signature.</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="al:IntSType" use="required" fixed="1">
                <annotation>
                    <documentation>Attribute used to specify compliance with XSD schema.</documentation>
                </annotation>
            </attribute>
        </complexType>
    </element>

    <complexType name="EinvoiceChangeStatusRequestHeaderType">
        <attribute name="UUID" type="al: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" use="required" type="al:UTCSType">
            <annotation>
                <documentation>Element represents date and time of sending the request message to the CIS. </documentation>
            </annotation>
        </attribute>
    </complexType>
    <complexType name="EinvoiceChangeStatusResponseHeaderType">
        <attribute name="UUID" type="al: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="al: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="al:UTCSType" use="required">
            <annotation>
                <documentation>Element represents date and time of sending the response message to the TCR. </documentation>
            </annotation>
        </attribute>
    </complexType>
    <complexType name="RegisterEinvoiceRequestHeaderType">
        <attribute name="UUID" type="al: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" use="required" type="al:UTCSType">
            <annotation>
                <documentation>Element represents date and time of sending the request message to the CIS. </documentation>
            </annotation>
        </attribute>
    </complexType>
    <complexType name="RegisterEinvoiceResponseHeaderType">
        <attribute name="UUID" type="al: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="al: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="al:UTCSType" use="required">
            <annotation>
                <documentation>Element represents date and time of sending the response message to the TCR. </documentation>
            </annotation>
        </attribute>
    </complexType>
    <complexType name="GetEinvoicesRequestHeaderType">
        <attribute name="UUID" type="al: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" use="required" type="al:UTCSType">
            <annotation>
                <documentation>Element represents date and time of sending the request message to the CIS. </documentation>
            </annotation>
        </attribute>
    </complexType>
    <complexType name="GetEinvoicesResponseHeaderType">
        <attribute name="UUID" type="al: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="al: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="al:UTCSType" use="required">
            <annotation>
                <documentation>Element represents date and time of sending the response message to the TCR. </documentation>
            </annotation>
        </attribute>
    </complexType>
    <complexType name="GetTaxpayersRequestHeaderType">
        <attribute name="UUID" type="al: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" use="required" type="al:UTCSType">
            <annotation>
                <documentation>Element represents date and time of sending the request message to the CIS. </documentation>
            </annotation>
        </attribute>
    </complexType>
    <complexType name="GetTaxpayersResponseHeaderType">
        <attribute name="UUID" type="al: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="al: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="al:UTCSType" use="required">
            <annotation>
                <documentation>Element represents date and time of sending the response message to the TCR. </documentation>
            </annotation>
        </attribute>
    </complexType>
    <complexType name="EinvoiceType">
        <sequence>
            <element name="Pdf" type="base64Binary" minOccurs="0" maxOccurs="1">
                <annotation>
                    <documentation>Element contains base64 Einvoice pdf.</documentation>
                </annotation>
            </element>
        </sequence>
        <attribute name="EIC" type="al:UUIDSType" use="required" >
            <annotation>
                <documentation>Einvoice EIC.</documentation>
            </annotation>
        </attribute>
        <attribute name="DocNumber" type="al:String100SType" use="required" >
            <annotation>
                <documentation>Document number.</documentation>
            </annotation>
        </attribute>
        <attribute name="DocType" type="al:DocumentType" use="required" >
            <annotation>
                <documentation>Document type.</documentation>
            </annotation>
        </attribute>
        <attribute name="RecDateTime" type="al:UTCSType" use="required" >
            <annotation>
                <documentation>Einvoice receive date time.</documentation>
            </annotation>
        </attribute>
        <attribute name="DueDateTime" type="al:UTCSType" use="required" >
            <annotation>
                <documentation>Einvoice due date time.</documentation>
            </annotation>
        </attribute>
        <attribute name="Status" type="al:EinvoiceStatusType" use="required" >
            <annotation>
                <documentation>Einvoice status</documentation>
            </annotation>
        </attribute>
        <attribute name="Amount" type="al:DecimalSType" use="required" >
            <annotation>
                <documentation>Einvoice amount.</documentation>
            </annotation>
        </attribute>
        <attribute name="PartyType" type="al:PartyTypeSType" use="required">
            <annotation>
                <documentation>Party type.</documentation>
            </annotation>
        </attribute>
    </complexType>
    <complexType name="EinvoicesType">
```

```
        <sequence>
            <element name="Einvoice" type="al:EinvoiceType" minOccurs="0" maxOccurs="100">
                <annotation>
                    <documentation>Element representing a single Einovice.</documentation>
                </annotation>
            </element>
        </sequence>
    </complexType>
    <complexType name="TaxpayerType">
        <attribute name="Name" type="al:String100SType" use="required" >
            <annotation>
                <documentation>Taxpayer name.</documentation>
            </annotation>
        </attribute>
        <attribute name="Tin" type="al:NUISType" use="required" >
            <annotation>
                <documentation>Taxpayer NUIS</documentation>
            </annotation>
        </attribute>
        <attribute name="Address" type="al:String100SType" use="optional" >
            <annotation>
                <documentation>Taxpayer address</documentation>
            </annotation>
        </attribute>
        <attribute name="Town" type="al:String100SType" use="optional" >
            <annotation>
                <documentation>Taxpayer town</documentation>
            </annotation>
        </attribute>
        <attribute name="Country" type="al:CountryCodeSType" use="optional">
            <annotation>
                <documentation>Taxpayer country.</documentation>
            </annotation>
        </attribute>
    </complexType>
    <complexType name="TaxpayersType">
        <sequence>
            <element name="Taxpayer" type="al:TaxpayerType" minOccurs="0" maxOccurs="100">
                <annotation>
                    <documentation>Element representing a single taxpayer.</documentation>
                </annotation>
            </element>
        </sequence>
    </complexType>
    <complexType name="EinvoiceEnvelopeType">
        <choice minOccurs="1" maxOccurs="1">
              <element name="UblInvoice" type="base64Binary">
                <annotation>
                    <documentation>Element contains base64 encoded UBL Invoice XML.</documentation>
                </annotation>
              </element>
              <element name="UblCreditNote" type="base64Binary">
                <annotation>
                    <documentation>Element contains base64 encoded UBL CreditNote XML.</documentation>
                </annotation>
              </element>
            <element name="CiiInvoice" type="base64Binary">
                <annotation>
                    <documentation>Element contains base64 encoded CII Invoice XML.</documentation>
                </annotation>
            </element>
        </choice>
    </complexType>
    <complexType name="UblEnvelopeType">
        <sequence>
            <any processContents="skip" minOccurs="1" maxOccurs="1"/>
          </sequence>
    </complexType>
    <complexType name="CiiEnvelopeType">
        <sequence>
            <any processContents="skip" minOccurs="1" maxOccurs="1"/>
          </sequence>
    </complexType>
    <complexType name="WarningsType">
        <sequence>
            <element name="Warning" type="al:WarningType" minOccurs="1" maxOccurs="200">
                <annotation>
                    <documentation>Element representing an warning in list of warnings.</documentation>
                </annotation>
            </element>
        </sequence>
    </complexType>
    <complexType name="WarningType" >
        <attribute name="Code" type="al:String20SType" >
            <annotation>
                <documentation>Warning code.</documentation>
            </annotation>
        </attribute>
        <attribute name="Description" type="al:String100SType" >
            <annotation>
                <documentation>Warning description.</documentation>
            </annotation>
        </attribute>
    </complexType>
    <complexType name="EICsType">
        <sequence>
            <element name="EIC" type="al:UUIDSType" minOccurs="1" maxOccurs="100">
                <annotation>
                    <documentation>Element representing a single einvoice id.</documentation>
                </annotation>
            </element>
        </sequence>
    </complexType>
    <complexType name="TaxpayerFilterType">
        <choice minOccurs="1" maxOccurs="1">
              <element name="Tin" type="al:NUISType">
                <annotation>
                    <documentation>Element contains filter for Nuis.</documentation>
                </annotation>
              </element>
              <element name="Name" type="al:String100Min3SType">
                <annotation>
                    <documentation>Element contains filter for name.</documentation>
                </annotation>
              </element>
        </choice>
    </complexType>

    <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="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="IntSType">
        <annotation>
            <documentation>Positive integer.</documentation>
        </annotation>
        <restriction base="int">
            <minExclusive value="0"/>
        </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="String100SType">
        <annotation>
            <documentation>String of up to 100 characters.</documentation>
        </annotation>
        <restriction base="string">
            <minLength value="1"/>
            <maxLength value="100"/>
        </restriction>
    </simpleType>
    <simpleType name="String100Min3SType">
        <annotation>
            <documentation>String of 3 up to 100 characters.</documentation>
        </annotation>
        <restriction base="string">
            <minLength value="3"/>
            <maxLength value="100"/>
        </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="NUISType">
        <annotation>
            <documentation>NUIS constructed in one letter - eight numbers - one letter pattern, unique.</documentation>
        </annotation>
        <restriction base="string">
            <pattern value="[a-zA-Z]{1}[0-9]{8}[a-zA-Z]{1}"/>
        </restriction>
    </simpleType>
    <simpleType name="EinvoiceStatusType">
        <annotation>
            <documentation>Einvoice status types.</documentation>
        </annotation>
        <restriction base="string">
            <enumeration value="ACCEPTED">
                <annotation>
                    <documentation>Accepted type.</documentation>
                </annotation>
            </enumeration>
```

```
            <enumeration value="REFUSED">
                <annotation>
                    <documentation>Refused type.</documentation>
                </annotation>
            </enumeration>
            <enumeration value="PAID">
                <annotation>
                    <documentation>Paid type.</documentation>
                </annotation>
            </enumeration>
            <enumeration value="PARTIALLY_PAID">
                <annotation>
                    <documentation>Partially paid type.</documentation>
                </annotation>
            </enumeration>
        </restriction>
    </simpleType>
    <simpleType name="DocumentType">
        <annotation>
            <documentation>Document code types.</documentation>
        </annotation>
        <restriction base="string">
            <enumeration value="80">
                <annotation>
                    <documentation>Debit note related to goods or services type.</documentation>
                </annotation>
            </enumeration>
            <enumeration value="82">
                <annotation>
                    <documentation>Metered services invoice type.</documentation>
                </annotation>
            </enumeration>
            <enumeration value="84">
                <annotation>
                    <documentation>Debit note related to financial adjustments type.</documentation>
                </annotation>
            </enumeration>
            <enumeration value="380">
                <annotation>
                    <documentation>Commercial invoicetype.</documentation>
                </annotation>
            </enumeration>
            <enumeration value="383">
                <annotation>
                    <documentation>Debit note type.</documentation>
                </annotation>
            </enumeration>
            <enumeration value="386">
                <annotation>
                    <documentation>Prepayment invoice type.</documentation>
                </annotation>
            </enumeration>
            <enumeration value="393">
                <annotation>
                    <documentation>Factored invoice type.</documentation>
                </annotation>
            </enumeration>
            <enumeration value="395">
                <annotation>
                    <documentation>Consignment invoice type.</documentation>
                </annotation>
            </enumeration>
            <enumeration value="575">
                <annotation>
                    <documentation>Insurer's invoice type.</documentation>
                </annotation>
            </enumeration>
            <enumeration value="623">
                <annotation>
                    <documentation>Forwarder's invoice type.</documentation>
                </annotation>
            </enumeration>
            <enumeration value="780">
                <annotation>
                    <documentation>Freight invoice type.</documentation>
                </annotation>
            </enumeration>
        </restriction>
    </simpleType>
    <simpleType name="ResponseCodeType">
        <annotation>
            <documentation>Response code types.</documentation>
        </annotation>
        <restriction base="string">
            <enumeration value="ACCEPTED">
                <annotation>
                    <documentation>Accepted type.</documentation>
                </annotation>
            </enumeration>
        </restriction>
    </simpleType>
    <simpleType name="PartyTypeSType">
        <annotation>
            <documentation>Party types.</documentation>
        </annotation>
        <restriction base="string">
            <enumeration value="BUYER">
                <annotation>
```

```
                    <documentation>Buyer type.</documentation>
                </annotation>
            </enumeration>
            <enumeration value="SELLER">
                <annotation>
                    <documentation>Seller type.</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>

</schema>
```