Krajowy System e-Faktur

Interface specifications
National e-Invoice System (KSEF)

Ministry of Finance

December 13, 2023

Version 1.5

REGISTER OF CHANGES

Data	Version	Description
18.10.2021	1.0	Base version of the document.
20.10.2021	1.1	Editorial corrections
08.05.2023	1.2	Update
25.07.2023	1.3	New version of the schema
07.09.2023	1.4	Description of calls distinguishing the system environment
13.12.2023	1.5	Adding descriptions for verification links and invoice hiding services

Contents

1. Glossary of concepts and terms used

KSeF	National e-Invoice System
System	System KSeF, API Systemu KSeF
Context	The entity and its identifier to which all operations in the System concern. Each interactive session is established and batch shipments are performed on behalf of this entity. This is, for example, the entity issuing the invoice (or receiving in the case of self-invoicing).
API	Application Programming Interface
XML	Extensible Markup Language
XSD	XML Schema Definition
JSON	JavaScript Object Notation
PEM	Privacy Enhanced Mail – text-based key storage format cryptographic
THE	Distinguished Encoding Rules – binary key storage format cryptographic
TLS	Transport Layer Security
PKCS	Public-Key Cryptography Standards – definitions of cryptographic standards
RSA	Rivest–Shamir–Adleman – private-public key algorithm
AES	Advanced Encryption Standard – symmetric key algorithm
ECB	Electronic CodeBook
CBC	Cipher Block Chaining
PKCS1Padding	The name of the PKCS#1 cryptographic complement
PKCS5Padding	The name of the cryptographic complement of the PKCS#7 standard
SHA-256	Secure Hash Algorithms 256 bit – cryptographic hash function
Base64	A transport encoding format that allows for bytes to be written in 64 format printable characters
PZ	Trusted Profile
XAdES	XML Advanced Electronic Signatures – digital signature format
Enveloped	Signature Format – Surrounded
Envelping	Signature Format – Surrounding
AIDS	Object IDentifier
CRL	Certificate Revocation List
OCSP	Online Certificate Status Protocol

2. System environment.

The system has the following environments:

Production environment (prod) – the production environment of the system. Invoices issued in this environment are full-fledged documents and carry all legal consequences.

Base address of the **environment_path environment:** https://ksef.mf.gov.pl/

Test environment (test) – an environment intended for testing a new solution by interested entities involved in the development of invoicing software. Self-generated signatures and seals can be used in the test environment. Invoices issued in the test environment will not have any legal effects and will be deleted from the system after a specified period of time.

Base address of the **environment_path environment:** https://ksef-test.mf.gov.pl/

Pre-production environment (demo) – an environment intended for testing the new solution by interested entities involved in the development of invoicing software. The environment is based on actual credentials consistent with the business owner information registry. In order to log in to the service, you must have actual permissions, similar to those for the production environment. Invoices issued in the pre-production environment will not have any legal effects and will be deleted from the system after a specified period of time.

Base address of the **environment_path environment:** https://ksef-demo.mf.gov.pl/

3. Overview

Communication with the System is based on the Taxpayer's Context. For batch shipping, this is it taxpayer issuing invoices. For interactive operations, this may be an entity the invoice issuer, the entity receiving the invoices or an authorized entity.

The system consists of three areas:

- batch shipping, which is a set of operations and a process that allows you to issue many invoices at the same time
  %environment_path %/ openapi/ gtw/ svc/ api/ KSeF-batch.yaml
- general operations enabling access to the System without requiring authentication
  %environment_path %/ openapi/ gtw/ svc/ api/ KSeF-common.yaml
- interactive operations enabling, among others, credential management, fast shipping invoices or searching and accessing invoices.
  %environment_path %/ openapi/ gtw/ svc/ api/ KSeF-online.yaml

The system requires authentication in accordance with defined authentication vectors and is based on authorization consistent with the internal credential model.

4. Authentication

4.1 Overview

The authentication mechanism allows for verification of the identity of the entity trying to gain access to the System. Identity is based on qualified identity sources: qualified certificate and Trusted Profile.

4.2 Podpis XAdES
https://www.w3.org/TR/XAdES/

4.2.1 Otaczany
Transformaty
http://www.w3.org/TR/1999/REC-xpath-19991116 - not(ancestor-or-self::ds:Signature) http://www.w3.org/2002/06/xmldsig-filter2 http://www.w3.org/2000/09/xmldsig#enveloped-signature

4.2.2 Surrounding
Allowable Transforms http://www.w3.org/2000/09/xmldsig#base64

4.2.3 Detached
Format not allowed.

4.3. Authentication vectors 4.3.1
Asynchronous vectors
Authentication will only occur after proper verification of the certificate qualifications, and the delay is directly related to the CRL and OCSP mechanisms.

4.3.1.1 Qualified signature
Certificate confirmed by a qualified certification center (https://www.nccert.pl/).

Required entity attributes
OID.2.5.4.42
OID.2.5.4.4

Optional entity attributes
OID.2.5.4.5
OID.2.5.4.3

Unacceptable entity attributes
OID.2.5.4.97

Recognized patterns OID.2.5.4.5
(PNOPL|PESEL).*?(?<number>\\d{11})
(TINPL|NIP).*?(?<number>\\d{10})

4.3.1.2 Qualified seal Certificate
confirmed by a qualified certification center (https://www.nccert.pl/).

Required entity attributes
OID.2.5.4.97

Optional entity attributes
OID.2.5.4.3

Unacceptable OID entity
attributes.2.5.4.5
OID.2.5.4.42
OID.2.5.4.4

Acceptable patterns OID.2.5.4.97
(VATPL).*?(?<number>\\d{10})

4.3.1.3 Signature certificate fingerprint

SHA-256 hash (http://www.w3.org/2009/xmldsig11#dsa-sha256) certificate, allowing the use of qualified signature certificates that do not have the appropriate identifiers defined in the entity attribute OID.2.5.4.5.

4.3.2 Synchronous vectors

In this case, identity confirmation is implicit due to trust in the identity source system.

4.3.2.1.Trusted profile

XAdES signature with the seal of the Minister of Digitization containing the structure http://crd.gov.pl/xml/schematy/ppzp/ in the element xades:SignerRole/xades:ClaimedRoles/xades:ClaimedRole indicating the authenticated person.

4.3.2.2 Authorization token

An identifier generated in the System by an authenticated entity containing a subset of the authorizations of this entity. The token is returned only once when it is generated and from the moment of authentication of the assumer it can be used to authenticate and authorize the entity in the basic credential model.
The limitation of the use of a token is the roles of the authentication vector that was used to generate the token. The token's roles can only be a subset of the parent authentication vector's roles, and if the parent authentication vector loses a role, the same role (if it was assigned to the token) is disabled. If a previously lost role is re-assigned to the parent authorization vector, the token will also regain it (it will be enabled, if it had one before).

Tokens are not subject to updating, the only permissible operation after creating a token is its invalidation.

5. Authorization

5.1 Overview

The authorization mechanism allows for the provision of appropriate services to an authenticated entity in a selected context. The basis for authorization is the selected Context (NIP or internal identifier) and the authentication vector (signature, stamp, PZ, token).

5.2 Credential Model

A scheme that allows you to determine access to the Subject Context based on a defined network of connections.

5.2.1 Identifiers

Categorization of identifiers supported by the System.

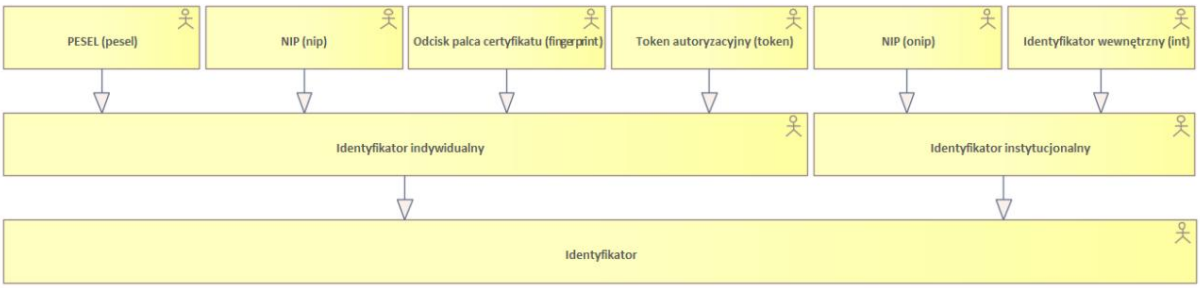

Figure 1 Identifiers

5.2.2 Relationships

Roles assigned to entities in the appropriate Contexts along with their effective date.
Links are valid until further notice.

The association consists of the Context in which it was granted, the indicated role defining the scope of the authorization, the entity identifier (NIP, PESEL) or authentication vector (authorization token, certificate fingerprint) to which the authorization was granted and the time stamp of the beginning of its validity.

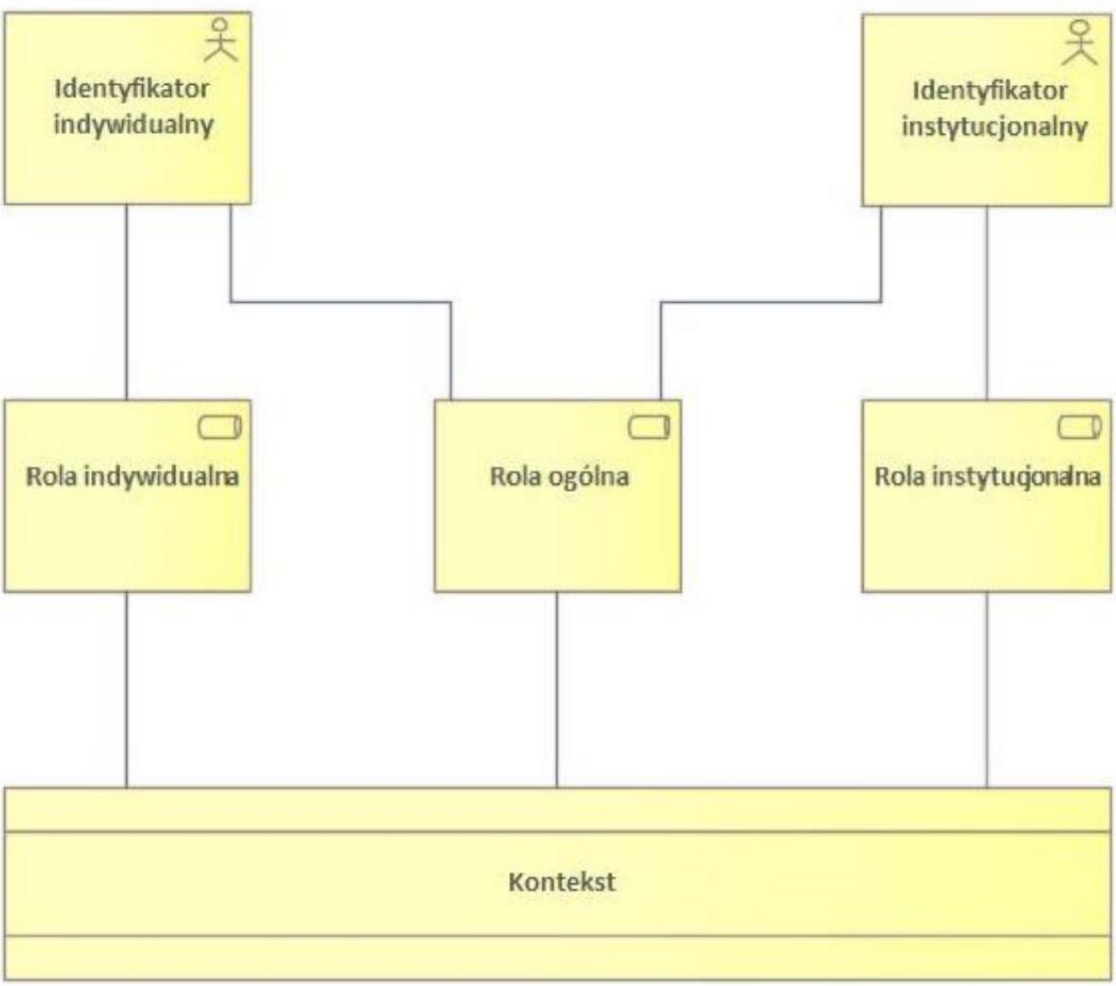

Figure 2 Connections

5.2.2.1 Static

Relationships from central systems, e.g. defining the entity owner.

5.2.2.2 Dynamic

Links provided in the System by authorized entities.

5.2.2.3 NIP-PESEL

Relationships allowing the interchangeable use of tax identification numbers (NIP) and PESEL (Personal Identification Number).
identifying the same person.

5.2.3 Role

Specific permissions in the System that can be assigned to entities in the appropriate context.

Figure 3 Roles

5.2.3.1 Overview

Owner – [owner] facade role, which is a set of all permissions granted by ***the invoice – read/ write*** and ***credentials – read/ manage roles.***

Invoices – reading/ writing – [invoice_read, invoice_write] operational roles, authorizing the functionality as indicated in the name – issuing and searching for invoices.

Credentials – reading/ managing – [credentials_read, credentials_manage] operational roles, authorizing the functionality as indicated in the name – searching and managing (giving and receiving) credentials.

Tax representative – [tax_representative] facade role, which is a set of all rights granted by the ***invoice roles – read/ write,*** authorizing the entity to perform the above operations on behalf of the entity granting the authorization. The role only works in tandem with **owner** or **invoice** role – **read/write.**

Self-invoicing – [self_invoicing] flag role, authorizing the entity to issue invoices on behalf of the entity granting the authorization. The role only works in conjunction with ***the Owner*** or ***Invoices - Entry role.***

Court bailiff – [court_bailiff] flag role, authorizing the entity to issue enforcement invoices. The role only works in tandem with the ***Enforcement Operations role.***

Enforcement authority – [enforcement_authority] flag role, authorizing the entity to issue enforcement invoices. The role only works in tandem with the ***Enforcement Operations role.***

Enforcement Operations – [enforcement_operations] an operational role that can only be assigned in the same context that has previously been flagged as ***bailiff*** or ***enforcement authority.***

Local Government Unit – parent – [local_government_unit] flagship role indicating the superior local government unit.

Local Government Unit – subordinate – [local_government_sub_unit] flagship role indicating a subordinate local government unit in the context of a superior unit local government.

VAT group – parent – [vat_group_unit] flag role indicating the parent unit of the VAT group.

VAT Group - Sub - [vat_group_sub_unit] flag role indicating the sub-unit of the group VAT in the context of the parent entity of the VAT group.

Subunit Management – [subunit_manage] operational role that can be assigned only in the same context that was previously flagged as **a Local Government Unit Territorial – parent** or **VAT Group – parent.**

5.2.3.2 Individual roles

Given by the administrator

Credentials – read/manage

Enforcement operations

Management of the subordinate entity

Broadcast by the System

Owner

5.2.3.3 Institutional roles

Given by the administrator

Self-billing

Tax representative

Awarded by the Office
Court bailiff
Enforcement authority
Local Government Unit – superior/subordinate
VAT group – parent/child

5.2.3.4 General roles

Given by the administrator
Invoices – read/write

5.3 Authentication with a qualified signature with a Tax Identification Number in the serial number or a qualified seal

In the case of authentication using the NIP identifier consistent with the declared Context, authorization takes place with the presumption of the owner role, omitting the enumeration of context roles.
If a tax identification number different from the declared Context is used, authorization assumes the role of the owner of the entity consistent with the authentication, but the context roles are fully calculated.

5.4 Authentication with a qualified signature with a PESEL number in the serial number, a trusted profile or a fingerprint of the signature certificate

In the case of authentication using a PESEL identifier or a fingerprint of a signature certificate, authorization takes place in accordance with the calculated context roles to the full extent.

5.5 Authentication with an authorization token

In the case of authentication using an authorization token, authorization occurs in accordance with enumerated context roles in a simplified scope (credentials only direct). An additional limitation is the roles of the authentication vector it generated token. The token's roles can only be a subset of the roles of the parent authentication vector, a if a role is lost by the parent authentication vector, the same role (if any assigned to the token) is disabled. In case of reassignment of a previously lost role parent authorization vector, the token will also recover it (it will be enabled, if he had one before).
Tokens are not subject to updating, the only allowed operation after creating a token is its annulment.

5.6 Authorization of operations

5.6.1 Establishing an interactive session (signature or token)

Establishing an interactive session is only possible for authentication vectors assigned any operational role or *Owner facade.*

5.6.2 Issuing an invoice (batch / interactive)

5.6.2.1 Standard

To issue a standard invoice, you must have the *Invoice Record* role or *Owner facade.*

5.6.2.2 Self-billed

To issue an invoice in the self-invoicing mode, the same rights are required as in the case of a standard invoice and the fact that the contextual entity has been given *the Self-invoicing* role by the entity acting as a seller.

5.6.2.3 Authorized Person

To issue an authorized invoice it is necessary to have
or role Enforcement operations or *Owner* facades and the fact of giving the entity context
by the Flag Office *Court Bailiff* or *Enforcement Authority*
or the role of *the Owner's invoice* or facade Record and the fact that the contextual entity has been given a role *Tax representative* by an entity acting as a seller.

5.6.3 Downloading an invoice

To download the original invoice by KSeF number, you must have the *Invoice Reader* role or *the Owner's facade.*

5.6.4 Session Status (Batch/Interactive)

5.6.4.1 General

No additional permissions are needed to check the overall session status by reference number. There is no need for an interactive session.

5.6.4.2 Interactive free

Interactively checking the status of any session by reference number does not require additional permissions. The method requires an active interactive session and is limited to only sessions in the same Context.

5.6.4.3 Interactive current

Interactive checking of the status of the current session does not require additional permissions. Requires an active interactive session.

5.6.5 Credentials

5.6.5.1 Generation of authorization token

Token generation requires any operational role or *Owner facade.*

5.6.5.2 Granting and revoking permissions

To grant permissions, you must have the *Credential Management* or *Child Management* role or *Owner facade.*

5.6.6 Inquiries

5.6.6.1 Credentials

To search for credentials, you must have *the Read Credentials, Manage Credentials,* or *Owner facade role.*

5.6.6.2 Invoices

To search and download invoice headers or originals, you must have the *Invoice Reader* role or *the Owner facade.*

5.6.7 Payments

5.6.7.1 Payment ID

To assign and read the payment ID, you must have the *Invoice Reader* role or *the Owner facade.*

6. Encryption

5.1 Overview

Communication is encrypted at one or two levels.

The first is channel-level encryption secured with the TLS protocol. This level is always active regardless of interface.
Additional content encryption based on the AES symmetric key and securing this key through encrypting it with the System's RSA public key *(%environment_path %/ security/ pem* or *%environment_path %/ security/ der).*
Additional encryption is mandatory for batch sending and for an interactive session is optional. However, if the interactive session was established with a cryptographic declaration documents must and will be sent (issuing invoices) and received (searching for originals). encrypted with the same symmetric key.

5.2 Symmetric key

The acceptable AES symmetric key encryption algorithm is AES/CBC/PKCS5Padding (PKCS#7).
The acceptable symmetric key is AES with a length of 256 bits supported by a random initialization vector of 16 bytes.

5.3 Public key

The acceptable RSA public key encryption algorithm is RSA/ECB/PKCS1Padding (PKCS#1).

5.4 Cryptographic Declaration

It is mandatory in the case of batch sending and optionally in the case of an interactive session to declare the cryptographic methods used. Additionally, the declaration must contain a byte array of the AES symmetric key encrypted with the RSA public key *(%environment_path %/ security/ pem* or *%environment_path %/ security/ der)* and encoded with the Base64 algorithm, and a byte array of the initialization vector encoded with the Base64 algorithm.

7. Protocols

7.1 Overview

A protocol is used to transfer data between client systems and the System
HTTP and the REST protocol based on it. The security of the transport layer of communication is based on the TLS protocol.

7.2 HTTP – REST

Communication takes place in the REST architecture, i.e. by sending stateless messages through a uniform interface: HTTP method + data related to it, to the service address specified in the API.

The HTTP method determines whether the API data is to be downloaded or searched (GET), modified, added or deleting (PUT, POST, DELETE) data. Services consume address path control parameters, query control parameters and data streams. Services along with HTTP response status can return formatted data in supported formats.

Example response statuses:

Code	Status	Description
200	OK	Request processing completed successfully
201	CREATED	Request processing completed successfully - a new resource has been created on the server side
202	ACCEPTED	Request processing completed successfully - content accepted for further processing
400	BAD REQUEST	Invalid request, or no data found based on the request parameters
401	UNAUTHORIZED	Unauthorized access

Code	Status	Description
404	NOT FOUND	The requested content was not found
429	TOO MANY REQUESTS	Request limit reached
500	INTERNAL SERVER ERROR	Internal System error

7.3 TLS

To ensure data security, the system enforces encryption of the connection using the TLS protocol, which is an extension of the SSL protocol. Trust in the system comes from public use, qualified certificate with which the System authorizes its domain and establishes an encrypted session.

8. Data format

8.1 Overview

The system uses XML and JSON text data formats and a binary data stream.

8.2 XML

XML (Extensible Markup Language) text format supported by the XSD meta-definition (XML Schema Definition allows you to transfer data in a systematic way.

The format additionally supports the XAdES signature format.

Application:

Invoice document

https://www.podatki.gov.pl/e-deklaracje/dokumentacja-it/struktury-dokumentow-xml/#ksef

Initialization of the batch shipping process (document *http://ksef.mf.gov.pl/schema/gtw/svc/batch/init/request/2021/10/01/0001/InitRequest)* *%environment_path %/schema/gtw/svc/batch/init/request/2021/10/01/0001/initRequest.xsd* Interactive session initialization (document *http://ksef.mf.gov.pl/schema/gtw/svc/online/auth/request/2021/10/01/0001/InitSessionTokenRequest* and *http://ksef.mf.gov.pl/schema/gtw/svc/online/auth/request/2021/10/01/0001/InitSessionSignedRequest)* *%environment_path %/schema/gtw/svc/online/auth/request/2021/10/01/0001/authRequest.xsd*

8.3 JSON

JSON (JavaScript Object Notation) text format.

The JSON structure consists of nested blocks enclosed by braces { ... } containing names and field values of represented objects.

Application:

General System I/O communication (excluding binary communication).
%environment_path %/openapi/gtw/svc/api/KSeF-batch.yaml
%environment_path %/openapi/gtw/svc/api/KSeF-common.yaml
%environment_path %/openapi/gtw/svc/api/KSeF-online.yaml

8.4 Binary data stream

A byte stream allowing the transfer of any information of any size.

Application:

Initiation of the batch shipping process (signed document
http:// ksef.mf.gov.pl/ schema/ gtw/ svc/ batch/ init/ request/ 2021/10/01/0001/ InitRequest)

%environment_path %/ api/ batch/ Init Initialize
the interactive session (document
http:// ksef.mf.gov.pl/ schema/ gtw/ svc/ online/ auth/ request/ 2021/10/01/0001/ InitSessionTokenRequest or signed
document
http:// ksef.mf.gov.pl/ schema/ gtw/ svc/ online/ auth/ request/ 2021/10/01/0001/ InitSessionSignedRequest)

%environment_path %/ api/ online/ Session/ InitSigned
%environment_path %/ api/ online/ Session/ InitToken Sending
'part' in batch shipping process (encrypted part of zip archive)
%environment_path %/ api/ batch/ Upload/ {ReferenceNumber}/ {PartName}
Downloading an invoice in an interactive session (invoice document)
%environment_path %/ api/ online/ Invoice/ Get/ {KSeFReferenceNumber}
Downloading search results for original invoices (encrypted part of the search result)
%environment_path %/
api/ online/ Query/ Invoice/ Async/ Fetch/ {QueryElementReferenceNumber}/ {PartElementReferenceN
umber}

9. Compression

9.1 Overview

Invoice packages subject to batch shipment and invoice packages resulting from searching for originals are first subject to packaging and compression. Currently, the accepted format is ZIP.

9.2 ZIP

The standard of compression and packaging in one.

Acceptable methods:
DEFLATE

10. Operations

10.1 Overview

Communication with the System takes place in two ways: synchronous and asynchronous. Hi
The response does not contain business information, only information about the start
asynchronous process and its handle.

Token generation is both synchronous (returns the token) and asynchronous (starts the process
token authentication and authorization).

The initialization of an interactive session is both synchronous (returns a session token) and
asynchronous, due to the authentication process.

10.2 Synchronous

Simple operations whose implementation does not require a complex process. If the operation requires
authentication and this has not yet occurred, an error will be returned.

Synchronous operations:
%environment_path %/ api/ online/ Session/ AuthorisationChallenge
%environment_path %/ api/ online/ Session/ InitSigned

%environment_path %/ api/ online/ Session/ InitToken
%environment_path %/ api/ online/ Session/ GenerateInternalIdentifier/ {inputDigitsSequence}
%environment_path %/ api/ online/ Credentials/ GenerateToken
%environment_path %/ api/ online/ Invoice/ Get/ {KSeFReferenceNumber}
%environment_path %/ api/ online/ Payment/ Identifier/ GetReferenceNumbers/ {PaymentIdentifier}
%environment_path %/ api/ online/ Payment/ Identifier/ Request
%environment_path %/ api/ online/ Query/ Credential/ Sync
%environment_path %/ api/ online/ Query/ Credential/ Context/ Sync
%environment_path %/ api/ online/ Query/ Invoice/ Sync
%environment_path %/ api/ online/ Invoice/ Visibility/ Hide
%environment_path %/ api/ online/ Invoice/ Visibility/ Show

10.3 Asynchronous

Asynchronous operations are processes initiated by calling the first method and
verified by the second method of checking the status. There are additional charges for batch shipping
methods of sending data along with signaling the end of this sending and in the case of queries
method of downloading results.

The status check is based on the asynchronous operation ID called
item reference number.

Asynchronous operations:
Batch shipping:
%environment_path %/ api/ batch/ Init
%environment_path %/ api/ batch/ Upload/ {ReferenceNumber}/ {PartName}
%environment_path %/ api/ batch/ Finish
%environment_path %/ api/ common/ Status/ {ReferenceNumber}

Interactive session:
%environment_path %/ api/ online/ Session/ InitSigned
%environment_path %/ api/ online/ Session/ InitToken
%environment_path %/ api/ online/ Session/ Status
%environment_path %/ api/ online/ Session/ Status/ {ReferenceNumber}
%environment_path %/ api/ online/ Session/ Terminate

Permission management:
%environment_path %/ api/ online/ Credentials/ GenerateToken
%environment_path %/ api/ online/ Credentials/ RevokeToken
%environment_path %/ api/ online/ Credentials/ Grant
%environment_path %/ api/ online/ Credentials/ Revoke
%environment_path %/ api/ online/ Credentials/ ContextGrant
%environment_path %/ api/ online/ Credentials/ ContextRevoke
%environment_path %/ api/ online/ Credentials/ Status/ {CredentialsElementReferenceNumber}

Invoice shipping:
%environment_path %/ api/ online/ Invoice/ Send
%environment_path %/ api/ online/ Invoice/ Status/ {InvoiceElementReferenceNumber}

Invoice search:
%environment_path %/ api/ online/ Query/ Invoice/ Async/ Init

%environment_path %/ api/ online/ Query/ Invoice/ Async/ Status/ {QueryElementReferenceNumber}
%environment_path
%/ api/ online/ Query/ Invoice/ Async/ Fetch/ {QueryElementReferenceNumber}/ {PartElementReferenceN
umber}

11. Batch shipping

11.1 Overview

Batch shipping is a set of operations and a process that allows you to issue multiple invoices simultaneously and bypassing the size limitation of the invoice document existing in the interface interactive. The assumption of the process is the atomicity of operations, all invoice documents must be present correct and be accepted, otherwise the entire package is rejected.

Requirements: selected authentication vector (excluding token).

Limitations: minimum one invoice document, maximum size of part of the package after encryption cannot exceed 50MB, the number of archive parts cannot exceed 100.

11.2 Preparation for shipment

Before actually initiating the batch shipping process, you must prepare:

- AES symmetric key
- Initialization vector
- AES symmetric key encrypted with the System's RSA public key
- Compressed invoice documents into one archive
- SHA-256 archive digest
- Divided binary archive into parts no larger than 50 MB (please note that 50 MB limit applies to encrypted items)
- Parts of the archive encrypted using a previously generated symmetric key AES and the initialization vector
- SHA-256 hash of each encrypted part of the archive

11.3 Initialization of shipment

%environment_path %/ openapi/ gtw/ svc/ api/ KSeF-batch.yaml#/ batch/ Init

First, you need to prepare the document *http:// ksef.mf.gov.pl/ schema/ gtw/ svc/ batch/ init/ request/ 2021/10/01/0001/ InitRequest (%environment_path %/ schema/ gtw/ svc/ batch/ init/ request/ 2021/10/01/0001/ initRequest.xsd)* i complete it with the information from the previous step. Additionally, in *the DocumentType* section, you must declare which version of the invoice schema we will use in this session (so far it was always version 1) (if we declare version 2, in this session it will be possible to send only invoice xml files compatible with the v2 schema to send invoices in v1, you should establish a second session with the declared *DocumentType* for v1).
Then the prepared document should be signed (https://www.w3.org/TR/XAdES/) selected authentication vector. Finally, the signed shipment initiation document should be sent to the System end responsible for initiating the batch shipment process *(%environment_path%/ api/ batch/ Init).*

In response, you will receive a reference number that will be used, among others, for: to check the process status and obtaining a UPO.

11.4 Proper Dispatch

%environment_path %/ openapi/ gtw/ svc/ api/ KSeF-batch.yaml#/ batch/ Upload/ {ReferenceNumber}/ {PartName}

After receiving the response, the operations from the previous step should be sent previously prepared encrypted parts of the archive to the appropriate addresses (indicated in the response of the previous operation, e.g. *https:// ksef.mf.gov.pl/ api/ batch/ Upload/ {ReferenceNumber}/ {PartName}).*

11.5 Completion of shipment

%environment_path %/ openapi/ gtw/ svc/ api/ KSeF-batch.yaml#/ batch/ Finish

After the sending of all encrypted parts of the archive has been successfully completed, the archive should be invoked an operation that signals the completion of the shipping process and will begin processing procesu *(%environment_path %/ api/ batch/ Finish).*

11.6 Shipping status

%environment_path %/ openapi/ gtw/ svc/ api/ KSeF-common.yaml#/ common/ Status/ {ReferenceNumber}

Using the reference number obtained in response to the batch shipment initialization possible is to check the status of the process, at what stage it is, and if the process has been completed positively, downloading the UPO (https://ksef.mf.gov.pl/api/common/Status/{ReferenceNumber}).

The UPO is returned in the format of a signed Base64-encoded XML document.

12. General operations

12.1 Overview

General operations enable access to the System for operations that do not require authentication or authorization, e.g. enabling checking the status, downloading an invoice or obtaining a UPO without the need to establish an interactive session.

12.2 Session Status (Batch/Interactive)

%environment_path %/ openapi/ gtw/ svc/ api/ KSeF-common.yaml#/ common/ Status/ {ReferenceNumber}

The service allows you to check the status of batch processing or the status of an interactive session and the stage they are at based on their reference number *(%environment_path %/ api/ common/ Status/ {ReferenceNumber}).*

Additionally, in version 1 of the response, if the process has been successfully completed or the interactive session has been completed and at least one invoice has been accepted during it, the operation results in UPO.

The UPO is returned in the format of a signed Base64-encoded XML document.

For version 3 of the response, if the process has been successfully completed or the interactive session has been completed and at least one invoice has been accepted during it, the service returns the reference number of the UPO element, after which we can download it as an XML document.

12.3 Downloading the UPO

%environment_path %/ openapi/ gtw/ svc/ api/ KSeF-common.yaml#/ common/ Upo/ {ReferenceNumber}/ {UpoReferenceNumber}

The service allows you to download UPOs according to the criteria provided by the service *(%environment_path %/ api/ common/ Status/ {ReferenceNumber}).*

12.4 Downloading an invoice

%environment_path %/ openapi/ gtw/ svc/ api/ KSeF-common.yaml#/ common/ Invoice/ KSeF

The service allows you to download an invoice anonymously (without the need to establish an interactive session) according to specific criteria.

13. Interactive session

13.1 Overview

The session and interactive interfaces provide tools for, among others: credential management, quick sending of invoices or searching and accessing invoices. Unlike shipping batch shipping, where a single incorrect invoice rejects the entire package, in the case of interactive shipping each invoice is treated individually. Closing the interactive session results in generating a UPO with a list of all invoices processed correctly and accepted.

13.2 Establishing an interactive session

An authenticated and authorized interactive session is the basis of interactive communication. IN In the case of asynchronous authentication vectors, the session is initially exclusive authorized, therefore the effect of any operations is delayed until positive authentication.

The session is established in the taxpayer's Context and it is not possible to change the Context during it session validity.

The authentication vector points to the proxy of the Context entity. In a special case qualified seal issued to a Context entity, the entity appears in its own name.

13.2.1 Authorization Challenge

%environment_path %/ openapi/ gtw/ svc/ api/ KSeF-online.yaml#/ online/ Session/ AuthorisationChallenge

The first step in the process of establishing an interactive session is receiving a challenge authorization for the declared Context. The token and timestamp of the challenge are there necessary in the next steps *(%environment_path %/ api/ online/ Session/ AuthorisationChallenge).*

13.2.2 By signature

%environment_path %/ openapi/ gtw/ svc/ api/ KSeF-online.yaml#/ online/ Session/ InitSigned

Requirement: response of the current authorization challenge, selected authentication vector

In this case, a document *http:// ksef.mf.gov.pl/ schema/ gtw/ svc/ online/ auth/ request/ 2021/10/01/0001/ InitSessionSignedRequest (%environment_path %/ schema/ gtw/ svc/ online/ auth/ request/ 2021/10/01/0001/ authRequest.xsd)* should be supplemented with contextual information and the results obtained from triggering the challenge

authorization. The key is to choose the authorization type, which must be consistent with the selected vector authentication.

Additionally, i.e. in the case of a batch session, in the *DocumentType* section you must declare the version in which invoices will be sent.

The completed document should be signed with the selected authentication vector *(%environment_path %/ api/ online/ Session/ InitSigned).*

13.2.3 Tokenem

%environment_path %/ openapi/ gtw/ svc/ api/ KSeF-online.yaml#/ online/ Session/ InitToken Requirement: response of the current authorization challenge, authorization token obtained on based on the selected authentication vector

In this case, a document *http:// ksef.mf.gov.pl/ schema/ gtw/ svc/ online/ auth/ request/ 2021/10/01/0001/ InitSessionTokenRequest (%environment_path %/ schema/ gtw/ svc/ online/ auth/ request/ 2021/10/01/0001/ authRequest.xsd)* should be completed similarly to the previous case. The difference is that instead of type authorization, the Token field must be completed *(https:// ksef.mf.gov.pl/ api/ online/ Session/ InitToken).*

Additionally, i.e. in the case of a batch session, in the *DocumentType* section you must declare the version in which invoices will be sent.

The content of the Token field is a Base64-encoded byte array of a public-key encrypted string characters consisting of the concatenation of the authorization token, the separator character | and values numeric (long) timestamp of the authorization challenge (number of milliseconds since January 1 1970).

E.g.

Base64(encrypt(public_key, bytes(token + '|' + challengeTime)))

13.3 Session Status (Batch/Interactive)

Session status provides information about the current stage in the session process and invoices sent within it along with their sub-process stage.

Session status query operations support paging.

13.3.1 Interactive Free

%environment_path %/ openapi/ gtw/ svc/ api/ KSeF-online.yaml#/ online/ Session/ Status/ {ReferenceNumber}

Requirement: interactive session token, session reference number to validate

The operation allows you to check the status of any session of the selected Context based on knowledge of its reference number. This applies to both interactive and batch, active and historical sessions

(%environment_path %/ api/ online/ Session/ Status/ {ReferenceNumber}).

13.3.2 Interactive Current

%environment_path %/ openapi/ gtw/ svc/ api/ KSeF-online.yaml#/ online/ Session/ Status

Requirement: interactive session token

The operation allows you to check the status of the current session in which it occurs
check *(%environment_path %/ api/ online/ Session/ Status).*

13.4 End of an interactive session

%environment_path %/ openapi/ gtw/ svc/ api/ KSeF-online.yaml#/ online/ Session/ Terminate

Requirement: interactive session token

The interactive session expires after a defined period of inactivity (120 minutes). It is possible
however, forcing the session to close. The session has just ended and no further sessions can be made
invoice documents, starts the process of issuing UPO
(%environment_path %/api/online/Session/Terminate).

13.5 Generating an internal identifier

%environment_path %/ openapi/ gtw/ svc/ api/ KSeF-
online.yaml#/ online/ Session/ GenerateInternalIdentifier/ {inputDigitsSequence}

Requirement: interactive session token

A service that allows you to generate an internal ID for the Tax ID in the context of which the session was established.
The generated ID can be used at *%environment_path %/ openapi/ gtw/ svc/ api/*
KSeF-online.yaml#/ online/ Credentials/ ContextGrant
%environment_path %/ openapi/ gtw/ svc/ api/ KSeF-online.yaml#/ online/ Credentials/ ContextRevoke or establishing
a session to the context of the generated ID.

13.6 Issuing an invoice

%environment_path %/ openapi/ gtw/ svc/ api/ KSeF-online.yaml#/ online/ Invoice/ Send

%environment_path %/ openapi/ gtw/ svc/ api/ KSeF-
online.yaml#/ online/ Invoice/ Status/ {InvoiceElementReferenceNumber}

Requirement: interactive session token, owner façade role, invoice entry operational role or
enforcement operations

Limitations: The size of an unencrypted invoice document cannot exceed 1 MB and after
encrypted, it cannot exceed 2 MB.

Issuing an invoice is an asynchronous process. After sending the document from the System, it returns
information about process initialization along with the element number
(%environment_path%/ api/ online/ Invoice/ Send). Using the item number is possible
checking the current stage of processing and the final status (invoice accepted
or rejected,
%environment_path %/ api/ online/ Invoice/ Status/ {InvoiceElementReferenceNumber}).

13.7 Downloading an invoice

%environment_path %/ openapi/ gtw/ svc/ api/ KSeF-
online.yaml#/ online/ Invoice/ Get/ {KSeFReferenceNumber}

Requirement: interactive session token, KSeF invoice number, owner façade role, operational role
reading invoices

The operation allows you to download any Context invoice based on its unique KSeF number
(%environment_path %/ api/ online/ Invoice/ Get/ {KSeFReferenceNumber}).

13.8 Credentials

13.8.1 Generacja tokena autoryzacyjnego

%environment_path %/ openapi/ gtw/ svc/ api/ KSeF-online.yaml#/ online/ Credentials/ GenerateToken %environment_path %/ openapi/ gtw/ svc/ api/ KSeF-online.yaml#/ online/ Credentials/ RevokeToken

Requirement: interactive session token, owner facade role, operational role read or write invoices, read or manage credentials, enforcement operations

The operation allows you to generate an authorization token. Such a token is associated with the authentication vector used to establish the session in which it was generated and may contain only a subset of the roles of this vector. The token can be generated during an authorized but not yet authenticated session, but it will be active only after this session has been properly authenticated *(%environment_path %/ api/ online/ Credentials/ GenerateToken).*

The authorization token is returned synchronously once in the response of the generate method and cannot be retrieved again.

The token authentication process is asynchronous and its status is available after querying for the item number. The limitation of this operation is the role of the credential manager, entities without this role can check the session authentication status (appropriately high stage number, *%environment_path %/ api/ online/ Credentials/ Status/ {CredentialsElementReferenceNumber}).*

The authorization token can be revoked on request *(%environment_path %/ api/ online/ Credentials/ RevokeToken).*

13.8.2 Nadawanie i odbieranie uprawnieÿ

%environment_path %/ openapi/ gtw/ svc/ api/ KSeF-online.yaml#/ online/ Credentials/ Grant %environment_path %/ openapi/ gtw/ svc/ api/ KSeF-online.yaml#/ online/ Credentials/ Revoke %environment_path %/ openapi/ gtw/ svc/ api/ KSeF-online.yaml#/ online/ Credentials/ Status/ {CredentialsElementReferenceNumber}

Requirement: interactive session token, owner façade role, credential management operational role

The operation allows you to assign and revoke selected roles to selected authentication vectors in the context in which the session supporting the operation is established *(%environment_path %/ api/ online/ Credentials/ Grant, %environment_path %/ api/ online/ Credentials/ Revoke).*

Credential management is an asynchronous operation, and checking the current stage of application processing is possible based on the element number *(%environment_path %/ api/ online/ Credentials/ Status/ {CredentialsElementReferenceNumber}).*

13.8.3 Nadawanie i odbieranie uprawnieÿ kontekstowych

%environment_path %/ openapi/ gtw/ svc/ api/ KSeF-online.yaml#/ online/ Credentials/ ContextGrant %environment_path %/ openapi/ gtw/ svc/ api/ KSeF-online.yaml#/ online/ Credentials/ ContextRevoke %environment_path %/ openapi/ gtw/ svc/ api/ KSeF-online.yaml#/ online/ Credentials/ Status/ {CredentialsElementReferenceNumber}

Requirement: interactive session token, facade role owner, operational role child management, flag role parent local government unit or GVat

The operation allows you to assign and receive the selected credential management role authentication vectors in the context of the selected slave entity.

Credential management is an asynchronous operation, and checking is the current stage application processing is possible based on the item number *(%environment_path %/ api/ online/ Credentials/ Status/ {CredentialsElementReferenceNumber}).*

13.9 Inquiries

13.9.1 Credentials

%environment_path %/ openapi/ gtw/ svc/ api/ KSeF-online.yaml#/ online/ Query/ Credential/ Sync

Requirement: interactive session token, façade role owner, operational role read credentials

The operation allows a synchronous search for the given credentials of the Context in which a session has been established *(%environment_path %/ api/ online/ Query/ Credential/ Sync).*

13.9.2 Credentials issued by the parent entity

%environment_path %/ openapi/ gtw/ svc/ api/ KSeF-online.yaml#/ online/ Query/ Credential/ Context/ Sync

Requirement: interactive session token, façade role owner, operational role slave management

The operation allows for synchronous searches for assigned credentials by the parent entity *(%environment_path %/ api/ online/ Query/ Credential/ Context/ Sync).*

13.9.3 Invoices

The search criteria are based on technical and business parameters. The business criteria are described details in the invoice document template
https://www.podatki.gov.pl/e-deklaracje/dokumentacja-it/struktury-dokumentow-xml/#ksef
In particular, *the subjectType parameter:*
subject1 – the search context is in the subject field of the first invoice document
subject2 – the search context is in the subject field of the second invoice document
subject3 – the search context is in the third party field of the invoice document
subjectAuthorized – the search context is in the authorized entity field
invoice document

13.9.3.1 Synchronous Headers

%environment_path %/ openapi/ gtw/ svc/ api/ KSeF-online.yaml#/ online/ Query/ Invoice/ Sync

Requirement: interactive session token, owner façade role, invoice reading operational role

The operation allows for synchronous searching for invoices of the Context in which it was created session. In response, it returns the invoice headers *(%environment_path %/ api/ online/ Query/ Invoice/ Sync).*

The operation supports paging.

13.9.3.2 Asynchroniczne oryginaÿy

%environment_path %/ openapi/ gtw/ svc/ api/ KSeF-online.yaml#/ online/ Query/ Invoice/ Async/ Init
%environment_path %/ openapi/ gtw/ svc/ api/ KSeF-online.yaml#/ online/ Query/ Invoice/ Async/ Status/ {QueryElementReferenceNumber}

%environment_path %/ openapi/ gtw/ svc/ api/ KSeF-online.yaml#/ online/ Query/ Invoice/ Async/ Fetch/ {QueryElementReferenceNumber}/ {PartElementRefe renceNumber}

Requirement: interactive session token, owner façade role, invoice reading operational role

Asynchronous invoice lookup operation The context in which the session was established. In response, it returns binary original invoices. The search sequence begins by initializing the search criteria *(%environment_path %/ api/ online/ Query/ Invoice/ Async/ Init).* Then, based on the query number, it is possible to check the current stage of query processing *(%environment_path %/ api/ online/ Query/ Invoice/ Async/ Status/ {QueryElementReferenceNumber}).*
If the search is successfully completed, information about the results packages will be made available. Packages expire after the defined time described in their status metric (120 minutes). If the interactive session was established with a defined encryption context, the resulting packages will be encrypted according to the declared context *(%environment_path %/ api/ online/ Query/ Invoice/ Async/ Fetch/ {QueryElementReferenceNumber}/ {PartElementReferenceN umber})*

13.10 Payments

13.10.1 Identyfikator pŷatnoŷci

%environment_path %/ openapi/ gtw/ svc/ api/ KSeF-online.yaml#/ online/ Payment/ Identifier/ Request
%environment_path %/ openapi/ gtw/ svc/ api/ KSeF-online.yaml#/ online/ Payment/ Identifier/ GetReferenceNumbers/ {PaymentIdentifier}

Requirement: interactive session token, owner façade role, invoice reading operational role

The payment identifier is a number that aggregates one or more KSeF invoice numbers of the same issuer and recipient pair. To generate an identifier, it is necessary to declare a list of KSeF invoice numbers, where the Context of the established session is indicated as the recipient, and if the request is consistent, a unique identifier will be returned *(%environment_path %/ api/ online/ Payment/ Identifier/ Request).*

The same identifier can be used in a session whose Context is the issuer of these invoices to obtain a list of KSeF numbers *(%environment_path %/ api/ online/ Payment/ Identifier/ GetReferenceNumbers/ {PaymentIdentifier}).*

If one contractor wants to pay the other collectively for more than one invoice, using the payment identifier mechanism he can link information about paid invoices with the transfer itself by including the appropriate identifier in the transfer title.

13.11 Hiding invoices

13.11.1 Hiding an invoice

%environment_path %/openapi/gtw/svc/api/KSeF-online.yaml#/online/Invoice/Visibility/Hide
%environment_path %/api/online/Invoice/Visibility/Hide

Requirement: interactive session token, owner façade role, invoice reading operational role

A synchronous operation that allows you to mark an invoice as hidden. A hidden invoice cannot be retrieved simultaneously with a non-hidden invoice in one request: *%environment_path* %/api/online/Query/Invoice/Sync,
%environment_path %/api/online/Query/Invoice/Async/Init.

To determine whether we want to download hidden invoices or not, the isHidden parameter is used:
true – for hidden invoices,
false – for non-hidden invoices,
default false

To hide an invoice, it is necessary to provide the KSeF number of the invoice and a justification for the decision to hide it.

13.11.2 Restoring an invoice from hiding

%environment_path %/openapi/gtw/svc/api/KSeF-online.yaml#/online/Invoice/Visibility/Show
%environment_path %/api/online/Invoice/Visibility/Show

Requirement: interactive session token, owner façade role, invoice reading operational role

A synchronous operation that allows you to unmark an invoice as hidden. The invoice is then restored to its original state before being hidden.

To restore an invoice from hiding, it is necessary to provide the KSeF number of the invoice and a justification for the decision to withdraw the hiding.

14. Error handling

14.1 Overview

If an error occurs, a generic response explaining the problem will be returned.

Depending on the reason for the error, the response will be returned with the appropriate http status:

- 400 – when a business error occurred, e.g. the sent request is incorrect, does not contain the required structures or in the case of requests requiring a signature, the document is unsigned or it was done incorrectly
- 500 – when an internal System error occurred
- 501 – when an unknown System error occurred

The error message contains information such as the name of the service in the context of which it occurred, the Context reference number (if available), service code, timestamp and details.

The service code is a globally unique identifier that is uniquely associated with the error received.

Details provide additional descriptive information about the bug itself (primarily details
business errors) and the internal code for the type of error that occurred.

When reporting an error, you must provide at least the service code and timestamp.

15. Processes

15.1 Overview

Processes are asynchronous operations. Each process is initialized by calling a method
an asynchronous operation directly or indirectly as a subprocess of another process.

The process is identified by a globally unique item reference number and consists of

stages, and each of these stages is assigned a 3-digit status (stage) code number.

It is possible to check the status and stage of the process. The check is carried out by calling the asynchronous operation status method based on the previously received one item reference number.

Possible statuses:

- <100 – 200) – 1** - initial code, the process is in progress or has been initiated but not not yet started
- <200 – 300) – 2** - terminal code – success – the process has been completed successfully
- <300 – 400) – 3** - operational code – the process is performing defined tasks
- <400 – 500) – 4** - terminal code – error – the process was terminated due to a business error occurred

15.2 Authentication Sub-Process

The subprocess responsible for performing asynchronous vector authentication and decrypting the symmetric key provided in the request.

The subprocess is initiated by the batch dispatch process and session initialization interactive. The result of the subprocess is used in all other processes i subprocesses.

Subprocess stages:

Name	Description	Start	Success	Mistake
Authenticate	Authentication	100	310	410
DecryptKey	Decrypt the provided key	310	200	415

15.3 Batch Shipment Processing

The process responsible for processing a batch shipment, allowing for the issuance of multiple items invoices simultaneously. The assumption of the process is the atomicity of operations and all invoice documents must be valid and accepted, otherwise the entire package is rejected.

The process is initiated by asynchronous batch dispatch operations.

A successfully completed process is issued by the UPO.

Process stages:

Name	Description	Start	Success	Mistake
PartsProvidedCheck	Verification of the correctness of delivered package elements	100	300	405
Authorise	Process authorization	300	310	410
Security	Verification of the results of the authentication subprocess	310	315	415
DecryptParts	Decrypting encrypted parts of the archive. Combining the	315	320	420
MergeParts	decrypted parts into a primary archive	320	325	425
DecompressPackage	Decompression of the original archive	325	330	430
ExportAndInitSP	Data export and initialization of processing subprocesses invoices	330	335	435
Invoice	Verification of the results of the invoice processing sub-process	335	340	440
GenerateUPO	Generating UPO	340	200	445

15.4 Interactive Session Handling Process

The process responsible for handling interactive operations. Unlike the shipping process batch, if you send an invoice document and it is rejected, the rejection effect document applies only to this document and not to the entire session. Remaining invoices left accepted and subsequent invoice documents that will be accepted remain accepted. Atomicity is limited to a single invoice document when in In the case of batch shipping, atomicity covered the entire package.

The process is initiated by asynchronous interactive session initialization operations.

The successfully completed process issues a UPO (if at least one UPO has been sent and accepted invoice document).

Process stages:

Name		Start	Success	Mistake
Authorise	Process	100	310	410
Security	authorization Verification of the results of the authentication	310	315	415
SessionInitCheck	subprocess Verification of the session state, supporting, among others: extinction inactive sessions	315	350	450
SessionEndCheck	Waiting for the session to end, regardless of the reason: idle time has elapsed or the appropriate method has been invoked	350	355	455
Invoice	Verification of the results of the invoice processing sub-process.	355	360	460
GenerateUPO	Generating UPO	360	200	465

15.5 Invoice Processing Sub-Process

The subprocess responsible for processing the invoice document, verifying it and ultimately accepting or rejecting it.

The subprocess is initiated by the batch shipping process and the interactive session process.

Subprocess stages:

Name	Description	Start	Success	Mistake
Authorise	Subprocess authorization	100	310	410
Security	Verification of the results of the authentication subprocess	310	315	415
Decrypt	Decryption of an encrypted invoice document	315	320	420
VerifyInvoiceSemantics	Verification of the semantics of the invoice document	320	325	425
VerifyInvoiceEssentials	Verification of the business assumptions of the invoice document	325	330	430
BeforeAccept	Waiting for the remaining invoice documents from the package batch	330	335	435
Accept	Invoice acceptance and KSeF number generation	335	340	440
ArchiveData	Archiving invoice data	340	200	445

15.6 Invoice Search Process

The process responsible for searching for invoices.

The process is initiated by asynchronous query operations.

Process stages:

Name	Description	Start	Success	Mistake
Authorise	Process authorization	100	310	410
Security	Verification of the results of the authentication subprocess	310	315	415
Statistics	Statistics analysis to optimize the query	315	320	420

Name	Description	Start	Success	Mistake
SplitDefineAndInitSP	Division into subqueries and initialization of subprocesses preparing part of the answer	320	325	425
Part	Verification of the results of part preparation subprocesses answers	325	330	430
ValidateResponse	Verification of response consistency	330	200	435

Subprocess stages:

Name	Description	Start	Success	Mistake
PreparePart	Executing the query and preparing the result	100	200	410

15.7 Credential Processing

The process responsible for processing credentials, granting and revoking permissions, and authenticating the authorization token.

The process is initiated by asynchronous credential operations.

Process stages:

Name	Description	Start	Success	Mistake
Authorise	Process authorization	100	310	410
Security	Verification of the results of the authentication subprocess	310	315	415
Process	Credential processing	315	200	420

16. Invoice verification and visualization

16.10 Introduction

Each invoice issued online, visualized in PDF form or as a traditional printout, should be provided with a verification link and its two-dimensional representation in the form of a QR code, under which the KSeF number should be located. Thanks to this marking, it will be possible to verify the presence of the invoice in the KSeF system.

16.11 Verification Links The

verification link should have the following format:

%environment_path %/web/common/verification/{ksefReferenceNumber}/{hash}

Where,

ksefReferenceNumber – Unique invoice number assigned by the KSeF system, (both 35 and 36 character KSeF numbers are accepted)

hash – Hash of the original invoice document in XML format, calculated based on the following algorithm:

Structure hash - urlEncode(Base64(SHA-256(xml)))

a) Calculate the invoice hash using the SHA-256 algorithm
b) Encode the previously obtained hash using the Base64 algorithm
c) Encode the previously encoded hash with the URL Encode percentage algorithm

Example:

a) Sha246 - 630b9c28b72cf3cba4ea2bcdd34fc2fcd45800a1f615db8e6f4bff71cc298d32
b) Base64(SHA-256(xml)) - YwucKLcs88uk6ivN00/C/NRYAKH2FduOb0v/ccwpjTI=
c) urlEncode(Base64(SHA-256(xml))-
YwucKLcs88uk6ivN00%2FC%2FNRYAKH2FduOb0v%2FccwpjTI%3D

The original invoice document sent by the API Client can be downloaded from KSeF as a byte stream, e.g. using the invoice download service in an interactive session:
%environment_path %/api/online/Invoice/Get/{KSeFReferenceNumber}

Examples of verification links:

%environment_path %/web/verify/4904089735-20220125-48BA3C-65D074-93/YwucKLcs88uk6ivN00%2FC%2FNRYAKH2FduOb0v%2FccwpjTI%3D

%environment_path %/web/verify/1111111111-20211231-62180B-218DB0-C0/jHbyhV1P8Yp4obWityeyYOLP3kcWu4IMi5fBJcbnElU%3D

%environment_path %/web/verify/1111111111-20211231-FDEBFB-FEC8EA-A4/4hDdWho%2FLmpXbC0Tsrx9RrIp8XAx%2FxKXMnmvE1narDU%3D

16.12 Kody QR

The graphical representation of the verification link in the form of a QR code should comply with the ISO/IEC 18004:2015 standard,

Error correction factor from L to H, for individual decision,

The size of the QR code on the printout and its exact location - an individual decision depending on the specific nature of the business,

The encoding type and version of the QR code can be determined automatically using available programming libraries to obtain the best readability of the QR code, with the desired size of the QR code on the printout.

An example of a QR code for the first sample link is as follows:

4904089735-20220125-48
BA3C-65D074-93