

NAV eVAT M2M system

eVAT M2M REST API

Interface & Developer's Documentation

TABLE OF CONTENTS

1	INTRODUCTION.....	8
1.1	Purpose.....	8
1.2	Conditions of use for taxpayers	8
1.3	Technical requirements for connecting.....	9
1.4	Technical requirements for management programs.....	10
2	DESCRIPTION OF THE EVAT REST API.....	11
2.1	Process of the service.....	12
2.2	General structure of XML messages.....	13
2.3	BasicEVatRequestType	13
2.3.1	BasicHeaderType	13
2.3.2	UserHeaderType.....	13
2.3.3	SoftwareType	13
2.4	BasicResponseType.....	13
2.5	Calculating requestSignature.....	13
2.6	Technical specifications of the service.....	13
2.6.1	General technical specifications	13
2.6.2	HTTP Headers	14
2.6.3	HTTP status codes	14
2.6.4	Size limit	14
2.6.5	Response time, timeout	14
2.6.6	Server clock, NTP	14
2.6.7	Maintenance method.....	15
2.6.8	Version Control	15
2.6.9	Character conversion	15
2.6.10	Traffic restriction	15
3	THE OPERATIONS AND OBJECTS OF EVAT M2M	16
3.1	XSD structure	16
3.2	Compiling the analytics XML.....	16
3.2.1	Declaration header data.....	17
3.2.2	Declaration statements	18

3.2.3	VAT analytics	19
3.2.4	Summary data of the original declaration.....	21
3.2.5	Summary data of the declaration.....	22
3.2.6	Supplementary sheets (optional)	23
3.2.7	Attachments (optional)	24
3.3	Business operations.....	24
3.3.1	/manageAttachmentUpload operation	25
3.3.2	/manageDeclarationUpload operation.....	26
3.3.3	/manageDeclarationPartition operation	30
3.3.4	/manageDeclarationFinalize operation	32
3.3.5	/manageDeclarationSubmission operation	33
3.3.6	/purgeAttachment operation	35
3.3.7	/queryAttachmentList operation.....	36
3.3.8	/queryCustomsDeclarationDigest operation	37
3.3.9	/queryCustomsDeclarationTaxCode operation	39
3.3.10	/queryDeclarationData operation	40
3.3.11	/queryDeclarationList operation	42
3.3.12	/queryDeclarationProcessingStatus operation.....	43
3.3.13	/queryDocumentList operation	45
3.3.14	/queryDocumentListResult operation	47
3.3.15	/queryInvoiceTaxCode operation	48
3.3.16	/queryTaxCodeCatalog operation	50
3.3.17	/queryVatDeclarationData operation.....	51
3.4	Clarifying information on how to fill out key fields.....	53
4	ERROR MANAGEMENT	64
4.1	General error codes	64
4.1.1	GeneralExceptionResponseType	64
4.1.2	GeneralErrorResponseType	64
4.2	Technical error codes	66
4.3	Validation error codes	73
4.3.1	Technical validation and blocking validation error codes	73

4.3.2	Warnings	73
5	ENVIRONMENTAL AVAILABILITY	74
5.1	User test environment.....	74
5.2	Live environment.....	74
6	HELPDESK AND TECHNICAL SUPPORT.....	76
6.1	Contact the Helpdesk	76
6.2	Availability on Github.....	76
6.2.1	Common repository	76
7	SYSTEM DIAGNOSTICS.....	77
8	ANNEXES.....	78
I.	WARNING MESSAGES SENT BY THE EVAT M2M SYSTEM	79
	ERROR type errors.....	79
	WARNING type messages	113
	INFO type messages	123

Terms & Abbreviations

Term	Description
Taxpayer	A taxpaying entity registered in Hungary who is obliged to file a VAT return.
API	Application Programming Interface.
API Gateway	The frontend performing the authentication, authorisation and distribution of requests.
VAT Act	Act CXXVII of 2007 on value added tax.
VAT analytics	A report for a given declaration period which contains the most important data of the base vouchers, as well as additional data (e.g. tax rate, tax code) which can be used to assign voucher data to the corresponding line of the current declaration form.
Declaration XML	<p>A data structure defined by NAV (NTCA) containing the tax settled, VAT analytics, and the data on the current VAT declaration forms.</p> <p>During the API communication, the taxpayer uploads the Declaration XML, from which eVAT M2M generates the declaration, which the taxpayer can approve after verification. Once approved, this XML legally fulfils the function of a declaration.</p>
Bevfeld XML	The XML file of the form-based declaration, the data content of which can be displayed in the General Form Filling Program (ÁNYK).
CDPS	Customs declarations processing system.
CDPS decision	The document that determines or provides information on the VAT due on imports.
M2M	Machine-machine connection.
NAV	The National Tax and Customs Authority.
SAF-T	A description of the structure of “Standard audit file – tax”, a file developed by the OECD for tax purposes. This also includes the data structure requirements of the record-keeping and accounting systems used by taxpayers. Between 2020 and 2021, NAV launched a pilot project on the practical use of the SAF-T_HU data structure with the involvement of customers. The results will be published on the https://github.com/nav-gov-hu/SAF-T_HU website.
SHA-512	A 512-bit Secure Hash Algorithm (Secure Hash Algorithm 3, RFC 6234).
SHA3-512	A 512-bit Secure Hash Algorithm with Keccak encryption (FIPS 202).
Standard tax code	A tax code file developed by NAV, which basically defines types of transactions.
taxpointDate	<p>The value of the taxpointDate field depends on the following:</p> <p>For vouchers associated to transactions that create a VAT payment obligation, the date on which the tax payment obligation arises under the VAT Act must be provided. This must fall within the tax settlement period.</p> <p>For vouchers associated to transactions that create a right of deduction, the date on which the right of tax deduction becomes available under the VAT Act must be provided. This date must fall within the tax settlement period or an earlier period, but not earlier than the period otherwise allowed for settlement by the VAT Act.</p>

	<p>With regard to the individual vouchers, it should be borne in mind that a voucher may be included in the tax settlements of multiple periods. An example of this is cash settlement, since in this case, multiple taxpointDate values are associated to a voucher.</p> <p>According to the above, the value of the taxpointDate can be clearly derived from the VAT Act, however, it is possible that this value is not stored by the taxpayer's accounting program. In this case, it is also a correct solution to set the value of the field to the last day of the tax settlement period.</p>
Technical User	A user required for data reporting via the REST API, which can be created by the primary user in the Online Invoice System.
XML	eXtensible Markup Language (W3C standard https://www.w3.org/TR/xml/).
XSD	XML Schema Definition file (W3C standard https://www.w3.org/TR/xmlschema11-1/).
REST	A Representational State Transfer (REST) web service, also called RESTful.
Primary User	The taxpayer, or their statutory or permanent representative: the user bearing with full access rights in the Online Invoice System. (The sole exception to this authority is that of data traffic via the REST API, which can only be performed via a technical user created by the Primary User.)
Endpoint	A path to access the service provided by the operation.
Operation	A collective term of information technology procedures and services that can be called via the offered REST web service.
Web Service	A collection of protocols and standards for inter-application data exchange.



Version History

Date	Author	Version	Change
05/05/2023	CsG, NA, HS, KTCs	0.1	Draft version intended for first review
07/06/2023	CsG	0.2	Version intended for first issue
12/06/2023	CsG	0.3	Error corrections
24/07/2023	CsG	0.4	Error corrections, update of Annex I
19/09/2023	CsG	0.5	earData.xsd DeclarationSummaryType modification; clarification of ERROR, WARN rules and error messages
30/01/2024	CsG	0.6	Removal of redundant XSD content, adding information about filling in supplementary sheets, adding error messages related to new validations, long run warnings
25/04/2024	RR, CsG	0.7	earData.xsd: VatDeclarationData/declarationInfo, addition of new fields related to self-auditing introduction of originalDeclarationSummary Adding fields to ManageDeclarationUploadRequest to manage and validate the XSD version Introduction of pre-separation checks for data related to self-auditing (ERROR type errors with no identifiers) Completion and correction of ERROR level errors for tests G0029, G0030, G0031, I0011 Creation of new ERROR level errors G0032, G0033, G0034, S0006, S0007, S0008, S0009, S0010, E0014, E0015, E0016, E0017
10/12/2024	CsG	0.8	Amendments concerning the legal relations between the predecessor and the successor entities. VAT changes for the year 2025 Withdrawal of examinations G0001, G0006, S0001, S0002
17/05/2025	FD	0.9	Legislative changes for 2025 Introduction of Annex 3 Modification of the DeclarationAdditionalDataType Establishing new ERROR-level faults with unique identifiers: F0008, S0011, S0012 Modification of new ERROR-level faults with unique identifiers: S0005

1 INTRODUCTION

1.1 Purpose

The primary objective of the document is to describe the submission of VAT analytics via an M2M interface, as well as the operation of connecting interfaces, to describe the XML message structures used for this purpose, and to support the integration of taxpayers' own accounting systems with the interfaces.

It enables taxpayers to import data from the tax authority system into their own systems that can be used for preparing the declaration, and to enable their management systems to submit the VAT declaration even using an M2M connection, i.e. without using the tax authority's system. By making the data content of the VAT declaration via computer more detailed, the taxpayer and NAV can both use the declaration as VAT analytics as well. The document therefore also defines the structure of VAT analytics, which is currently not regulated at the data level.

The use of an M2M connection can reduce the overall administrative burden for taxpayers, as no itemised VAT declarations (M forms) need to be filled in for declarations submitted in this way, since NAV has access to the full VAT analytics.

1.2 Conditions of use for taxpayers

The operation and design of the machine interface, the authentication and authorization mechanisms are based on the types defined in the common schema, and use the same authentication error codes as those NAV has already established in the Online Invoice System.

1) The VAT taxpayer, legal representative or permanent representative must log into the Online Invoice System at least once.

2) They must first create a technical user or set the appropriate permissions for an existing technical user on the user editor screen of the Online Invoice System. Data communication is only possible via an M2M connection via a technical user. The taxpayer can choose the number of technical users to create for M2M communication. A technical user can only be created by a primary user.

3) A signature key and a replacement key must be generated for the technical user. The signature key is required for calculating the *requestSignature* value, which is needed for signing messages; the replacement key is required for the server-side encoding and client-side decoding of the data transfer token.

4) Assignment of permissions

eVAT uses a multi-level permissions system. The first level is interface access privilege, while the second level contains the specific permissions of the eVAT interface, which also need to be set by primary users according to the operations they want to use. Without the appropriate permissions, the system will reject all requests from the client. Error codes are described in detail in the "**ERROR MANAGEMENT**" section.

4.1) When logged into the web-based Online Invoice System, primary users can use the *Users* menu to set the permissions for a given technical user, allowing them to access eVAT's machine interface.

4.2) the permissions applicable within the machine interface need to be set by switching to the eVAT web-based online system. On the online screens, the single sign-on (SSO) method allows users to authenticate only once.

The above-mentioned requirements apply per system level. Technical users and keys created and generated in the test environment cannot be used in the production environment, similarly to the Online Invoice System.

1.3 Technical requirements for connecting

- HTTPS – Secure HTTP
- Webservice
- REST API – REST interface required for the data reporting process
- XML – eXtensible Markup Language
- Encoding and encryption algorithms

In practice, the meaning of these expectations is that if a piece of management software was able to establish an API connection with the Online Invoice System, it should also be able to establish a connection with the eVAT M2M system, as they use the same technologies.

1.4 Technical requirements for management programs

- 1) The service interface is accessible to any management program capable of sending HTTP messages and compiling schema-compliant XML as defined in the specification.
- 2) The management program must also submit the authentication details for the technical user of the taxpayer upon each data submission and data query. The implementation required for this can be defined freely by the management program. It is not necessary to create client-side automations, processes can be designed freely according to taxpayer needs.
- 3) For successful authentication, the management program must implement the following encoding and encryption algorithms:
 - Base64 encode/decode (RFC 3548)
 - SHA-512 encode (RFC 6234)
 - SHA3-512 encode (FIPS 202)

The technical requirements for management programs are almost identical to those for Online Invoice API connections, as the basic processes are also almost identical to those developed for the invoice data service. Naturally, there are some differences, for example: this is not real-time data reporting.

2 DESCRIPTION OF THE eVAT REST API

The eVAT interface implements the following operations.

- manageAttachmentUpload: upload an attachment to a declaration
- manageDeclarationUpload: upload VAT analytics
- manageDeclarationPartition: upload VAT analytics partitions
- manageDeclarationFinalize: finalize the upload after the VAT analytics upload is complete
- queryDeclarationProcessingStatus: query the declaration processing result
- manageDeclarationSubmission: the interface for declaration submission
- queryCustomsDeclarationDigest: query of CDPS digest data
- queryInvoiceTaxCode: query of invoice tax codes and VAT relevant WARNs
- queryCustomsDeclarationTaxCode: query of CDPS decision tax codes
- queryDocumentList: start a voucher list query
- queryDocumentListResult: query the result of a voucher list query
- queryDeclarationList: query a list of declarations
- queryDeclarationData: download declaration data
- queryTaxCodeCatalog: query the tax code catalogue
- queryVatDeclarationData: download declaration analytics
- queryAttachmentList: list attachments
- purgeAttachment: delete an attachment

2.1 Process of the service

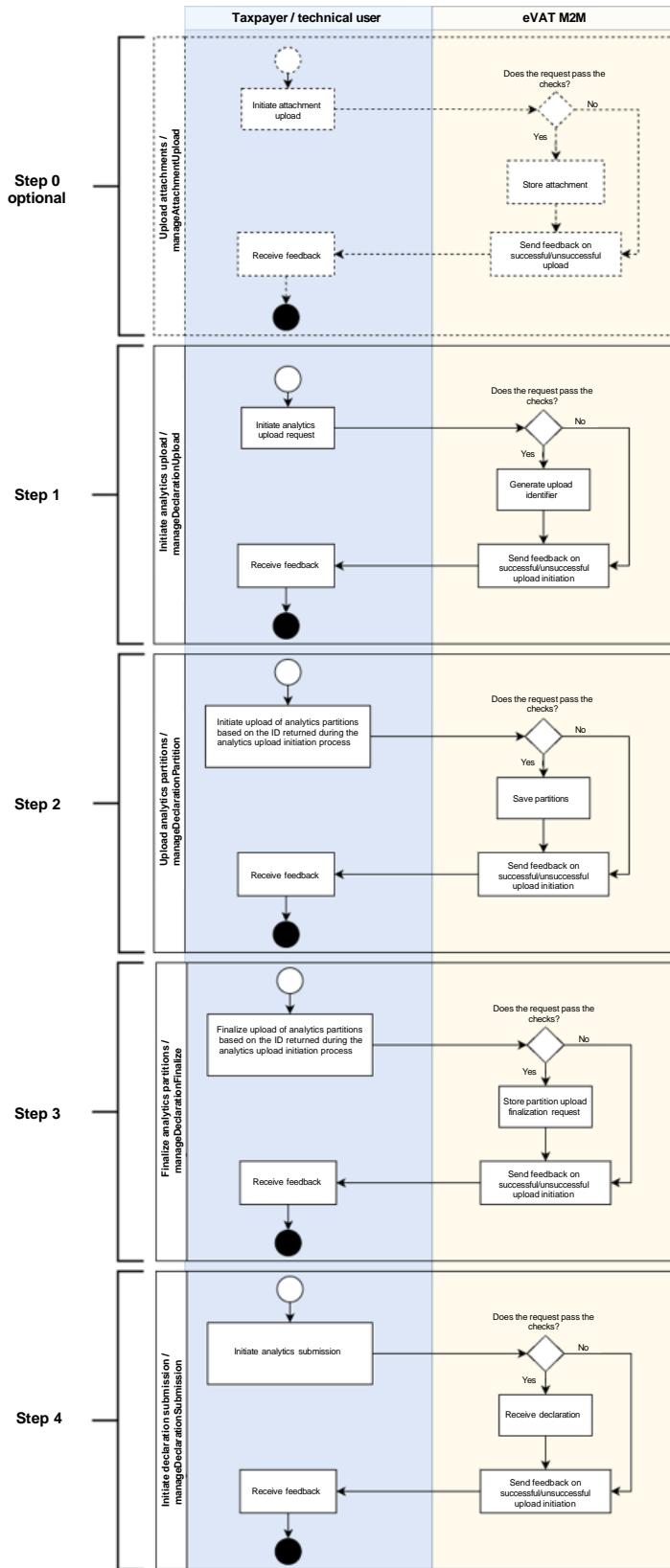


Figure 1 The service process of the M2M system

Upload attachments

- If there is an attachment to be attached to the declaration, upload should be started with this. If the declaration has no attachment, analytics submission should be started with the next step.

Initiate analytics upload

- The purpose of the service is to generate an upload identifier, referencing which analytics partitions can be uploaded. There may be only one valid upload ID at a given time, and when a new one is requested, the previous one is deleted immediately.

Upload analytics partitions

- The analytics partitions of the declaration can be submitted to NAV through this service. The size of a partition should not exceed 128MB

Finalize upload of analytics partitions

- Operation to finalize the submission of VAT analytics partitions. This is used to indicate that client-side partition uploads are complete and processing may begin

Initiate declaration submission

- Operation for submitting the VAT declaration. Only declarations with the FINISHED status can be submitted. Declarations may be submitted within the period set out in the applicable legislation.

2.2 General structure of XML messages

The eVAT interface has the number of root element pairs stipulated in the section “*Business operations*”. Some of the pairs are request type, while the others are response type, describing the request-response structure of the relevant operation.

2.3 BasicEVatRequestType

complexType **BasicEVatRequestType** is a mandatory part of each request element. The complex type header and user nodes are from common.xsd, while the software node type is described in earApi.xsd. The type header contains the technical data regarding the message exchange, the user part contains the data regarding authentication, while the software part contains the data of the program performing the operation.

2.3.1 BasicHeaderType

Within the requests, the header element is implemented by the *BasicHeaderType* element.

2.3.2 UserHeaderType

Within the requests, the user element is implemented by the *UserHeaderType* element.

2.3.3 SoftwareType

Within the requests, the software element is implemented by the *SoftwareType* element.

2.4 BasicResponseType

BasicResponseType is a mandatory part of every response element.

2.5 Calculating requestSignature

The method for calculating the requestSignature is described in section 2.4 of the Developer’s Documentation of the NAV API Gateway.

2.6 Technical specifications of the service

/analyticsService is a RESTful type stateless web service. The service has the following technical specifications.

2.6.1 General technical specifications

To the services, the proper XML request must be sent in the body, using the HTTP POST method, to which request the server sends a response with XML in the body. XML requests are expected to be in single-line format.

Exceptions to the XML request are file upload operations, which must be sent with the multipart/form-data media type, and an exception to XML response is the QueryDeclarationData operation, which also has a multipart/form-data response.

The requester defines the operation they wish to have executed by addressing the proper endpoint and assembling properly formatted, valid XML. Depending on whether or not the request is formatted correctly, the server will either send a business XML response, or just a standard HTTP response.

2.6.2 HTTP Headers

The request must specify the following HTTP header fields:

- content-type=application/xml
- accept=application/xml

Exceptions to this are the following operations with multipart request and/or response:

In the case of manageDeclarationPartition and manageAttachmentUpload, the content-type to be specified is:

- content-type = multipart/form-data

And for the queryVatDeclarationData operation, the value of the accept header should be:

- accept = multipart/form-data; application/xml

The save to database and the response will always be UTF-8, regardless of the encoding specified in the inquiry, therefore it is advised to use this type of encoding in the query.

2.6.3 HTTP status codes

For correct requests, the service will always return a HTTP 200 response. This does not necessarily indicate that the business execution of the content of the request was successful, it merely indicates that the request was correctly formatted, and the resource addressed was able to read and validate it. As the error codes managed by the service are mapped, the error codes returned are also considered successful responses. This means a HTTP 200 response may also contain a message with error codes.

For an explanation of responses for incorrect requests and other technical errors, please consult the error code chart in the section “**Error management**”.

2.6.4 Size limit

The HTTP POST body sent to the service cannot exceed 10 MB for operations other than file upload.

Exceptions to this are file upload operations with the multipart/form-data content-type.

When uploading a file, the maximum size of the declaration partition to be uploaded is 128 MB for the manageDeclarationPartition operation, and the maximum size of the attachment to be uploaded is 100 MB for the manageAttachmentUpload operation, based on the size of the octet-stream in the request.

2.6.5 Response time, timeout

The typical response time of the server is under 200 ms. The blocking timeout value for synchronous requests is 5000 ms. Please only consider response times to be timeouts on the client side if they exceed the above values.

The absolute timeout value is 60 sec. Failure to respond to an operation due to a 60-second timeout does not indicate an operation failure.

2.6.6 Server clock, NTP

The server receives time settings from a closed NTP server that is not accessible to the outside world. On the client side, synchronisation with server time is not a requirement, however, optionally, the following time synchronisation is possible: <http://www.pool.ntp.org/zone/hu> (connection requires an NTP client).

2.6.6.1 Converting local time to UTC

The generation of the correct client-side requestSignature value requires that the local time be converted to UTC. This can be done by adding or subtracting from local time in the client-side time zone, the number of hours that particular time zone is offset from UTC mean time. For time zones divided into summer and winter time (daylight saving), take this into account for the addition and subtraction as well.

2.6.7 Maintenance method

In maintenance mode, the system returns the error indicated in the “**Error Management**” section.

By design, it returns an HTTP 503 error.

2.6.8 Version Control

For service-related purposes, the version is defined in the URL, while the version for the business data model is defined by the value of the requestVersion tag in the HTTP body.

Major versions are defined as the versions between which the backward compatibility of the business data model cannot be guaranteed. Minor versions are versions within a single major version where compatibility is retained for business data.

New major versions will always have a new URL and a new XML namespace. Minor versions will inherit the URL and namespace data of their major versions.

2.6.9 Character conversion

Under no circumstances will server-side character conversion be performed on the submitted data.

2.6.10 Traffic restriction

Traffic restriction (Request Rate Limiting) can be expected on all interfaces of the M2M system. Clients that exceed the allowed limit will receive an HTTP 429 message as described in the section “**Error Management**”.

If the limit is exceeded, the transfer of requests may be resumed after the time window has expired.

3 THE OPERATIONS AND OBJECTS OF eVAT M2M

3.1 XSD structure

The element nodes listed in the tables below are the key complex nodes of the service. They typically contain several atomic elements and complex type nodes, to allow for the flexible assembly of request-response messages used by the operations.

The following XSDs are required to use eVAT M2M.

XSD name	Namespace	Description	Access
earAPI.xsd	http://schemas.nav.gov.hu/EAR/1.0/api	Contains the objects needed for calling the eVAT M2M XML API.	link
earBase.xsd	http://schemas.nav.gov.hu/EAR/1.0/base	Defines the basic business types that are specific to the eVAT system, but occur more frequently.	link
earData.xsd	http://schemas.nav.gov.hu/EAR/1.0/data	Schema containing the main business input to the eVAT API. It describes the declaration and the VAT analytics, along with the necessary supplementary sheets and attachments.	link
common.xsd	http://schemas.nav.gov.hu/NTCA/1.0/common	Common import for all NAV schemas. A schema describing atomic and derived types, the request-response structure of basic API communication, and generic error types.	link

3.2 Compiling the analytics XML

The taxpayer compiles the declaration XML in their management program, which contains:

- the (generic) header data of the declaration
- (optionally) declaration statements
- complete and itemised VAT analysis of the period
- (optionally) summary data of the original declaration
- summary data of the declaration
- (optionally) supplementary sheets
- (optionally) attachment data

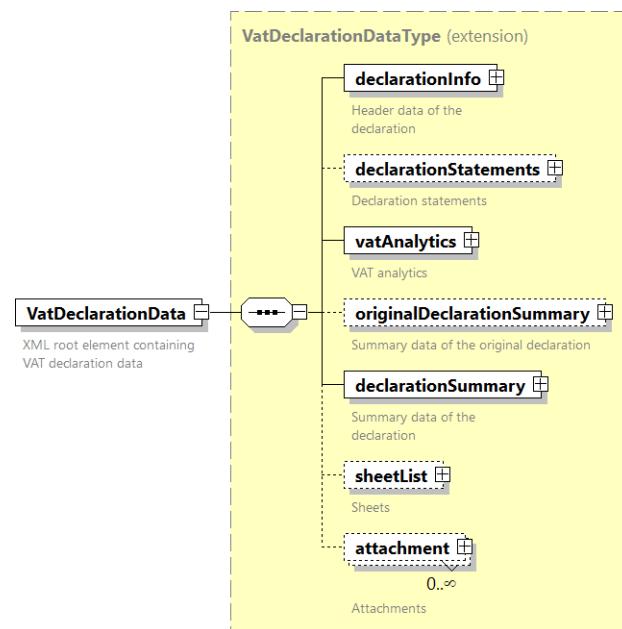
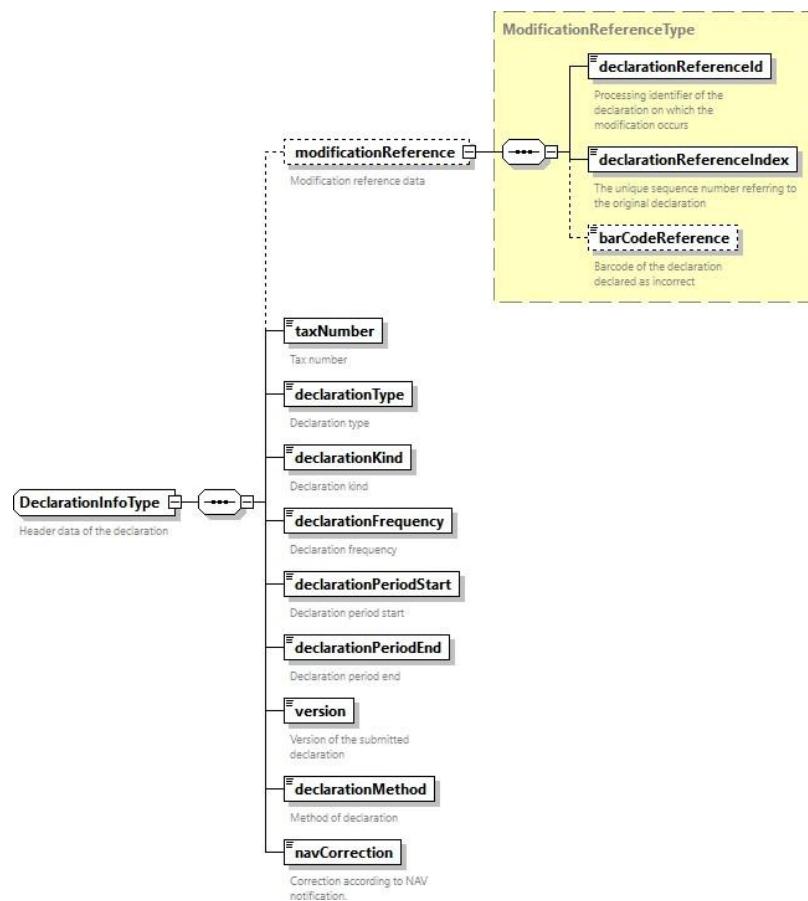


Figure 2 The structure of VatDeclarationData

3.2.1 Declaration header data



3. Figure The structure of DeclarationInfoType

Information on how to fill the object

The *modificationReference* node is used to reference amending declarations (self-audits, corrigendums, corrections) within the system boundary. The *modificationReference* tag can be populated when self-auditing, correcting or amending a previously submitted declaration (with the appropriate status). When submitting a basic declaration, the node cannot be populated.

It ensures uniqueness together with the rest of the head data. This uniqueness describes the rule that there can only be one submitted declaration, using the following:

- the processing identifier of the modified declaration (*declarationReferenceId*)
- an index signifying the order of the modification (*declarationReferenceIndex*)
- the barcode of the declaration identified as incorrect (*barCodeReference*)
- a given taxpayer (*taxNumber*)
- in a given declaration type (*declarationType* and *declarationKind* together)
- at a given declaration frequency (*declarationFrequency*)
- for a given period (*declarationPeriodStart* and *declarationPeriodEnd* together)
- with a given version number (*version*)
- with the declaration method (*declarationMethod*)
- with a flag showing a correction made at NAV's request (*navCorrection*)

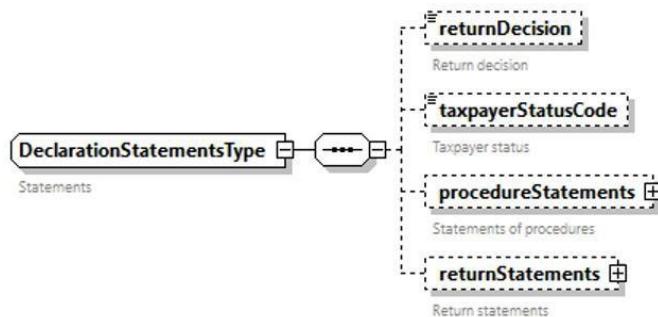
If a new version needs to be submitted via the API, the *version* value should be incremented. The value of *version* can only be a positive integer, monotonously increasing from 1.

Self-audits, corrigendums and corrections can only be submitted for declarations that have been successfully submitted and have the SUBMITTED status. The *declarationReferenceId* field must reference the previous declaration submitted by the taxpayer. The *declarationReferenceIndex* value is used to specify the order of the declarations submitted for self-audits, corrigendums and corrections of the basic declaration (that was filed first). The value of *declarationReferenceIndex* is a positive integer, monotonically increasing from 1.

The declaration type can be a self-audit or a corrigendum. Basic declarations can have self-audits and corrigendums, self-audits can have self-audits and corrigendums, while corrigendums can have corrigendums and self-audits. After a successfully submitted self-audit and corrigendum, it is also possible to submit additional self-audits and corrigendums.

When submitting a correction, the identifier sent by NAV must be entered into the *barCodeReference* field. The type of declaration must match the type of the (previously successfully submitted) declaration requiring correction.

3.2.2 Declaration statements



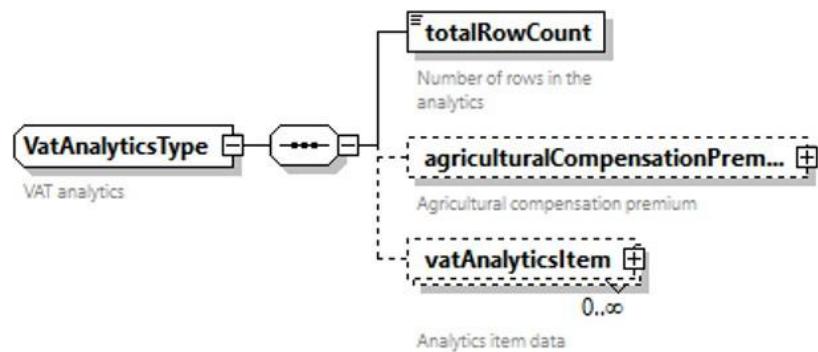
4. Figure The structure of *DeclarationStatementsType*

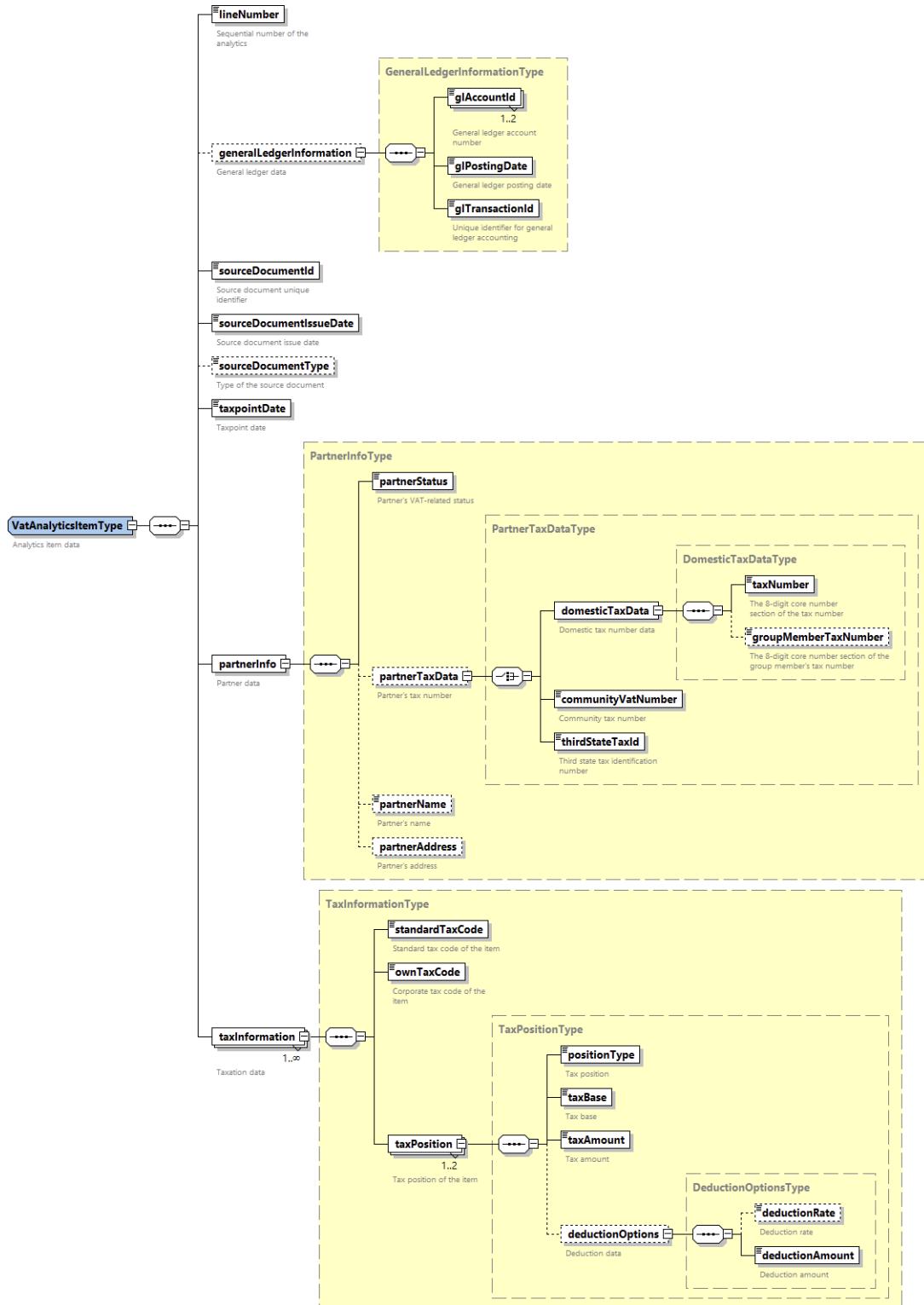
Information on how to fill the object

The *DeclarationStatementsType* node is used to specify various statements (form block C) and return provisions (form block D).

- It is not required to provide *returnDecision* in all declarations. Its value can be *NO_RETURN* (no return requested), *FULL_RETURN* (full return) or *TAX_ACCOUNT_TRANSFER* (transfer request).
- *returnStatements* and *returnMethod* can only be provided if *returnDecision != NO_RETURN*, but if this is the case, generating them is mandatory.
- Other statements can be generated by the taxpayer, taking into account the procedural and other rules of declaration.

3.2.3 VAT analytics





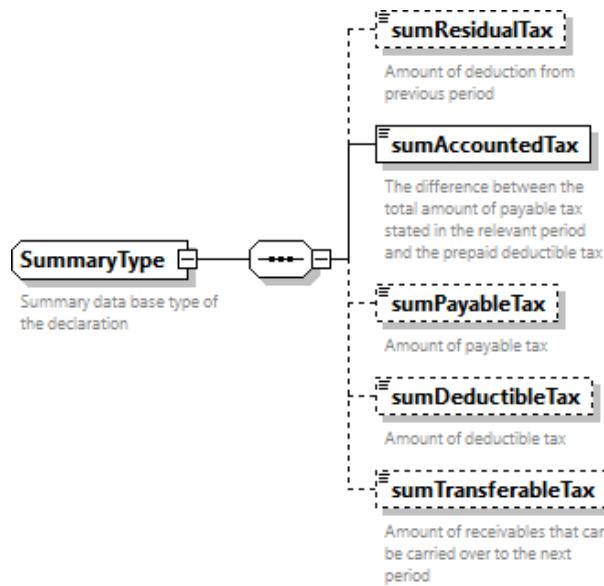
5. Figure The structure of VatAnalyticsItemType

Information on how to fill the object

- The VAT analytics node is mandatory in all declarations. The number of line items in the analytics should also be provided in all cases. If the taxpayer submits a null declaration (if there were no transactions generating tax obligations and no right of tax deduction is exercised during the period, in which case there are no meaningful analytics), this can be described with the expression totalRowCount = 0, and by omitting the node containing the analytics items (vatAnalyticsItem).
- The taxpayer's software numbers the VAT analytics items in the XML (lineNumber). The serial number can only consist of positive integers, monotonously increasing from 1 to infinity. No gaps or duplications in the numbering are allowed.
- The VAT analytics can optionally include general ledger data (general ledger invoice number, general ledger submission date, unique general ledger accounting identifier) in the generalLedgerInformation node. If any data is filled in, all data in the node must be filled in.
- VAT analytics is document-based and therefore must include the identifiers of the underlying vouchers (sourceDocumentId) and their issue dates (sourceDocumentIssueDate). The voucher identifier is multiform: it can be an invoice number, a receipt number, or a CDPS decision number, depending on the data source. Vouchers that exist only on the taxpayer's side (e.g. deeds of acquisition, contracts, etc.) should also be included here.
- The taxpointDate is a new concept alongside tax codes. The field must clearly indicate the date on which the tax payment obligation and/or right of deduction arises.
- For vouchers that have a relevant partner, this must be specified in the *partnerInfo* node. The structure is very similar to OSA, with more validation to enforce correct completion.
- Multiple standard tax codes may be associated to a given voucher. In addition to the standard tax codes, the taxpayer may also provide their own internal tax codes.
- A given standard tax code list can have at least one and at most two tax positions. From these declaration positions, at most one may be of payment (*positionType* = PAYABLE), at most one of deduction (*positionType* = DEDUCTIBLE) and at most one of other type (*positionType* = OTHER). No other type of tax position may be entered together with any other position.
- A given tax position must have a tax base (*taxBase*) and a tax amount (*taxAmount*). If the tax position is of deduction type, it may optionally have deduction data associated. Validation ensures that the deduction data are generated in the right place.
- The field of the VAT analytics containing all amounts should contain the unaggregated, raw values (not the amounts rounded to the nearest thousand are provided here as on the declaration form).

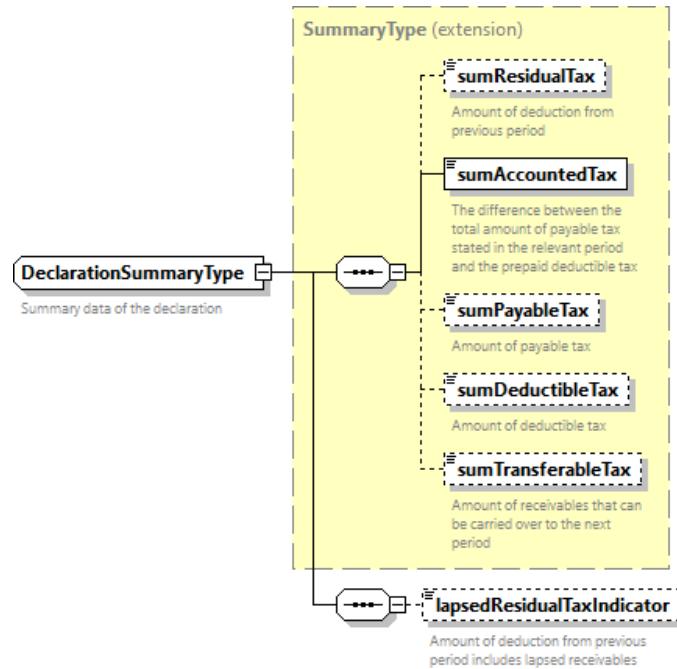
3.2.4 Summary data of the original declaration

When submitting a self-audit, corrigendum or correction, the taxpayer must provide the “main” figures from the original declaration. Based on the summary rows for the original and the current declaration, the system determines whether there has been an increase or decrease in payment obligations, then checks whether the supplementary sheets have been filled out accordingly. “Main” figures should be provided rounded to the nearest thousand forints. When rounding, the mathematical rounding rules should be applied.



6. Figure The structure of SummaryType, originalDeclarationSummary

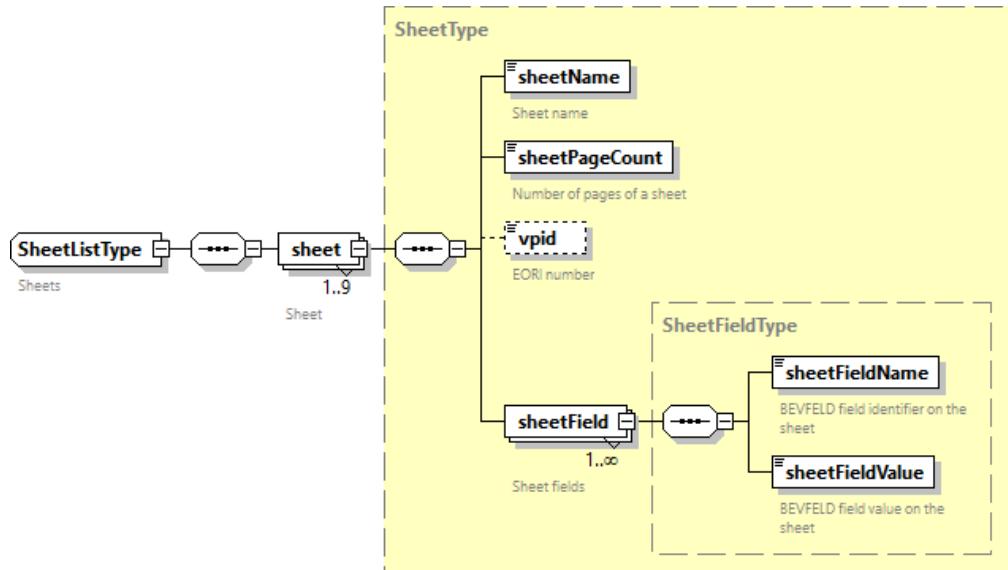
3.2.5 Summary data of the declaration



7. Figure The structure of DeclarationSummaryType

In addition to the analytics, the taxpayer must provide the “main” figures of the declaration. “Main” figures should be provided rounded to the nearest thousand forints. When rounding, the mathematical rounding rules should be applied. The system will perform *ERROR* validations concerning the correctness of these numbers.

3.2.6 Supplementary sheets (optional)



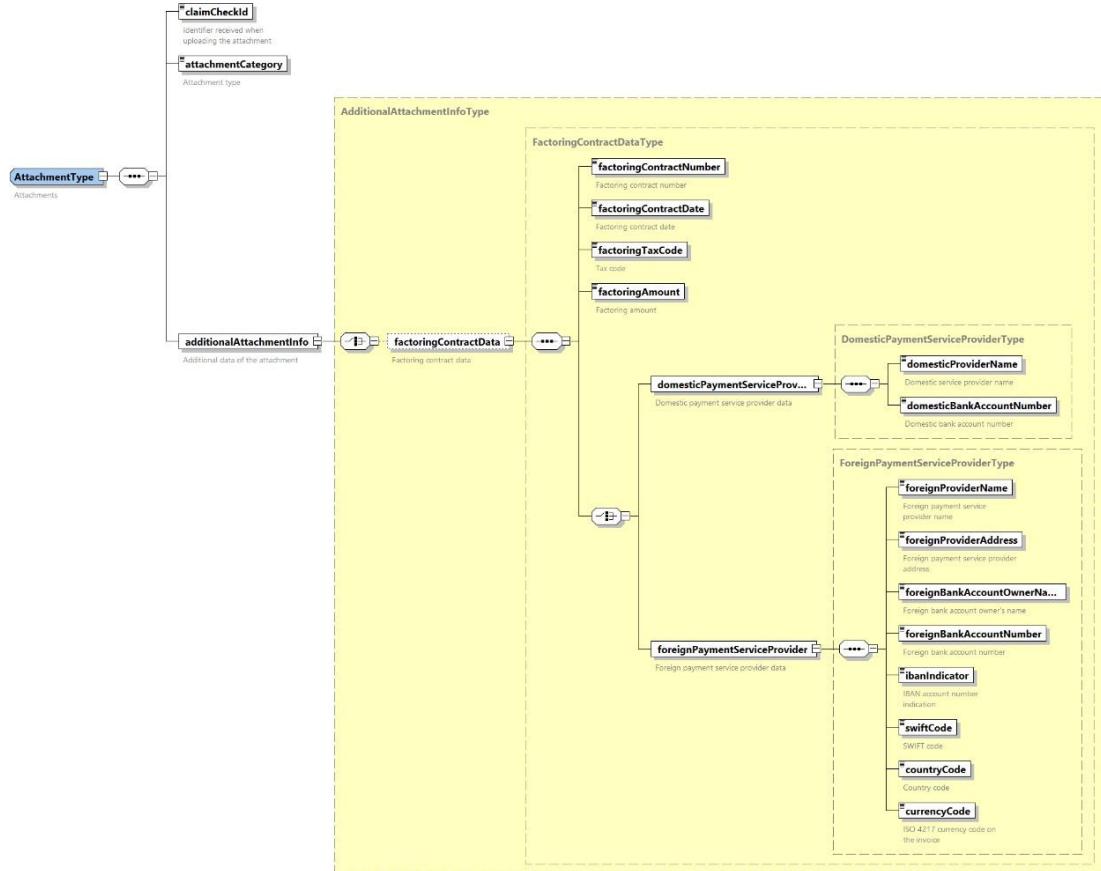
The declaration may have supplementary sheets attached. A total of 9 separate supplementary sheet types are handled by the eVAT system API, which are the following:

- Supplementary sheet 2
- Supplementary sheet 4
- Supplementary sheet 6
- Supplementary sheet 7
- Supplementary sheet 8
- Supplementary sheet 9
- Supplementary sheet A88
- Supplementary sheet 170
- Supplementary sheet EUNY

Most of these are generated depending on certain tax codes or the filling of payment provisions, and can thus be validated during processing.

The structure of the supplementary sheets follows the XML format of the General Form Filling Program (ÁNYK), and can be described by a key-value pair (*sheetFieldName* + *sheetFieldValue*). The *sheetFieldName* field will contain the standard common identifier (E.azon in the General Form Filling Program) and the *sheetFieldValue* will contain the value to be submitted. The exact field identifiers can be found on the published draft xx65A forms, or by downloading the corresponding form in the EAzon column of the csv file generated using the “Service/For Developers/Create field definition file (.CSV)” option in the General Form Filling Program (ÁNYK).

3.2.7 Attachments (optional)



8. Figure The structure of AttachmentType

Information on how to fill the object

The declaration may contain attachments. The attached file itself will be referenced in the declaration as *claimCheckId*. The validity of the *claimCheckId* is 72 hours, which means that attachments can be attached to a declaration within 72 hours of upload, in unlimited numbers. The advantage of this is that when a declaration runs into an error during asynchronous processing, neither the client nor the server side has to take into account the repeated transmission of the attachment (maximum 100MB) through the network. Currently, only a factoring contract can be attached to the VAT declaration. *attachmentCategory* and *additionalAttachmentInfo* have been created for the generic description of this. The former contains an attachment type indication, and the latter contains any metadata associated with the types. The internal structure of *additionalAttachmentInfo* is single-choice, i.e. only 1 type of element can be selected, even if it contains multiple types. The server will check based on the *attachmentCategory* value whether the attachment needs to be validated, and if so, which *additionalAttachmentInfo* node needs to be specified. This method allows for future expansion in a generic manner.

3.3 Business operations

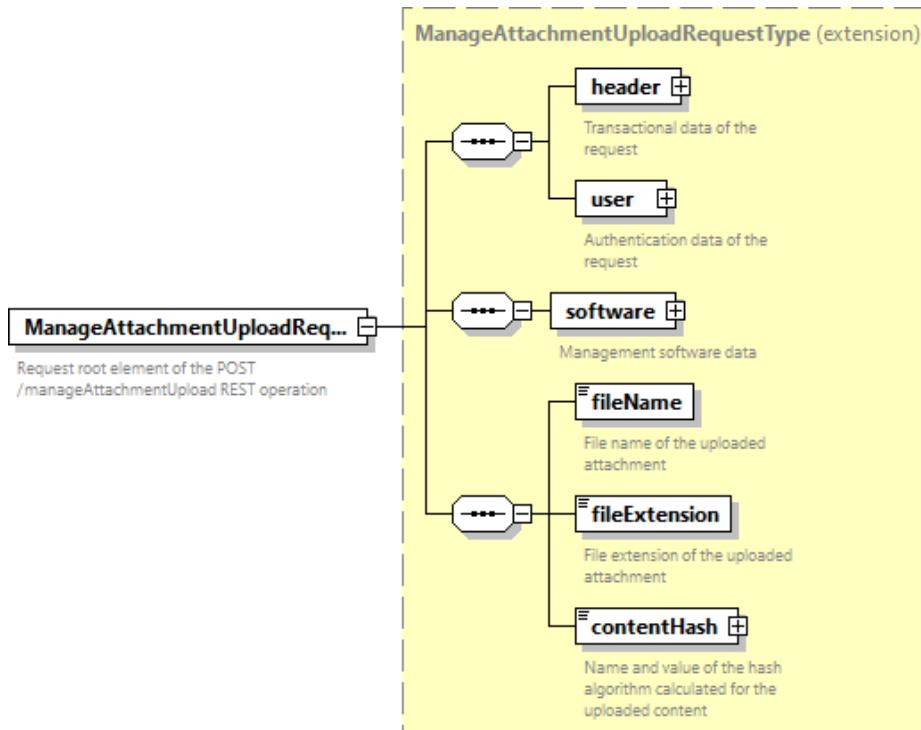
This section details the technical specifications of the implementation of service interfaces, and presents the root elements describing the various operations and request-response structures.

3.3.1 /manageAttachmentUpload operation

The operation is used to upload attachments optional attached to the declaration.

No additional permission other than authentication is required to upload the attachment.

element **ManageAttachmentUploadRequest**

<p><i>Diagram</i></p>  <pre> classDiagram class ManageAttachmentUploadRequestType { header user software fileName fileExtension contentHash } class ManageAttachmentUploadRequest { <> } ManageAttachmentUploadRequest < -- ManageAttachmentUploadRequestType </pre> <p>ManageAttachmentUploadRequestType (extension)</p> <ul style="list-style-type: none"> header [+] Transactional data of the request user [+] Authentication data of the request software [+] Management software data fileName [=] File name of the uploaded attachment fileExtension [=] File extension of the uploaded attachment contentHash [+] Name and value of the hash algorithm calculated for the uploaded content <p>ManageAttachmentUploadReq... [+] Request root element of the POST /manageAttachmentUpload REST operation</p>	<p><i>Namespace</i></p> <p>http://schemas.nav.gov.hu/EAR/1.0/api</p> <p><i>Type</i></p> <p>ManageAttachmentUploadRequestType extension</p> <p><i>Features</i></p> <p>content complex</p> <p><i>Tags</i></p> <p>header user software fileName fileExtension contentHash</p> <p><i>Content of</i></p> <p>Root element of the POST /manageAttachmentUpload REST operation request</p>
--	--

element **ManageAttachmentUploadResponse**

<i>Diagram</i>	<pre> classDiagram class ManageAttachmentRes... { <<Response root element of the POST /manageAttachmentUpload REST operation>> } class ManageAttachmentUploadResponseType { <<extension>> } ManageAttachmentRes... "3..>" ManageAttachmentUploadResponseType ManageAttachmentUploadResponseType "1..>" header ManageAttachmentUploadResponseType "1..>" result ManageAttachmentUploadResponseType "1..>" claimCheckId ManageAttachmentUploadResponseType "1..>" claimCheckIdValidFrom ManageAttachmentUploadResponseType "1..>" claimCheckIdValidTo header "1..>" Transactional data of the response result "1..>" Basic result data claimCheckId "1..>" Identifier received when uploading the attachment claimCheckIdValidFrom "1..>" Start of validity of the upload identifier claimCheckIdValidTo "1..>" End of validity of the upload identifier </pre>
<i>Namespace</i>	http://schemas.nav.gov.hu/EAR/1.0/api
<i>Type</i>	<u>ManageAttachmentUploadResponseType</u> extension
<i>Features</i>	content complex
<i>Tags</i>	<u>header</u> <u>result</u> <u>claimCheckId</u> <u>claimCheckIdValidFrom</u> <u>claimCheckIdValidTo</u>
<i>Content of</i>	Root element of the POST /manageAttachmentUpload REST operation response

Definition and related requirements

1. The maximum size of the attachment uploaded by the taxpayer is 100MB.
2. The maximum number of files the taxpayer can upload is 12.
3. The hash function is SHA3-512; if a different one is received in the *contentHash.cryptoType* field, the request returns an error.
4. The extension/type of the uploaded file can be PDF, JPEG or PNG. If the data contents imply a different data type when checking the file, the request returns an error.
5. The validity of the returned *claimCheckId* is 3 days (72 hours) from the time of issue. It should be ensured that the *claimCheckId* specified in the analytics does not expire until the */manageDeclarationFinalize* call.
6. Uploaded files are counted towards the quantity quota until submission (the */manageDeclarationSubmission* call).
7. If the uploaded attachment no longer exists when the declaration is submitted (e.g. the 72 hours have expired or it has been deleted), the declaration returns an ERROR when it is submitted.

3.3.2 /manageDeclarationUpload operation

The purpose of the operation is for the M2M system to generate an upload identifier for the technical user, referencing which analytics partitions can be uploaded. There may be only one valid upload ID (*declarationUploadId*) at a given time, and when a new one is requested, the previous one is deleted immediately. The upload ID is an ID bound to the taxpayer, so it can be used by multiple technical users.

Required permission: VM2M Declaration management

element **ManageDeclarationUploadRequest**

The structure of the /manageDeclarationUpload operation request is contained within the *ManageDeclarationUploadRequest* element.

The type is derived from *BasicEVatRequestType*, so in addition to the elements included in this, it contains a list containing the partitions of the analytics file to be uploaded, the hash of the entire file (not split into partitions), the type of the declaration, optionally the identifiers of the attachments attached to the declaration, as well as the XSD version (data required for XSD version management), the start of the declaration period, and the end of the declaration period.

When starting the upload of the declaration, the user must specify the declaration period and the version of the XSD (EARDATA.XSD) used.

Providing this information allows the taxpayer to submit analytics with an XSD version corresponding to the period of the basic declaration, self-audit, corrigendum and correction. The version and validity of the XSD will be verified based on the data provided.

The eVAT M2M system also stores information on which declaration periods it accepts analytics for, based on the XSD version. Currently, the system acknowledges the XSD versions shown in the table below:

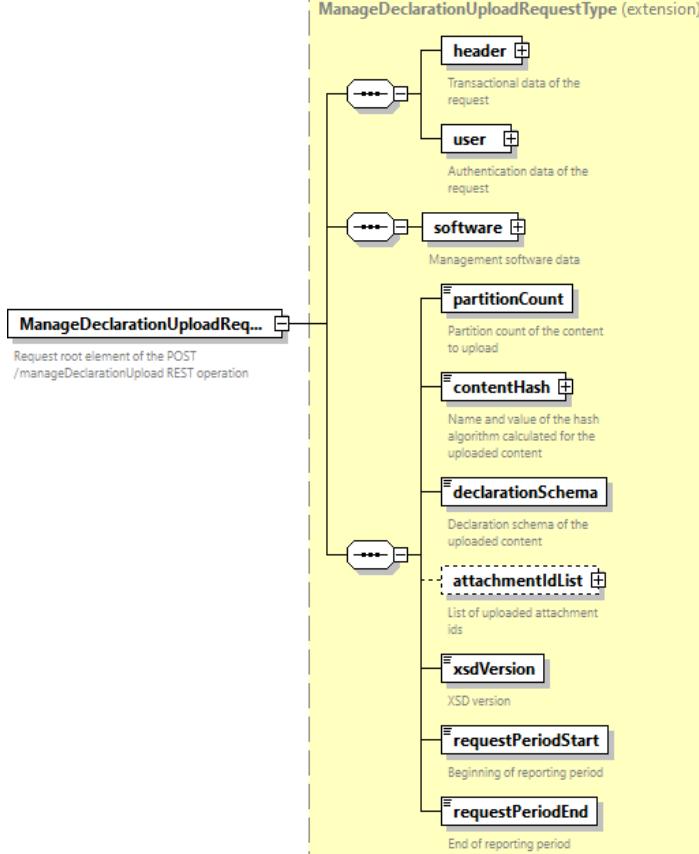
XSD version	Description		Start of XSD validity	End of XSD validity	Start of assigned declaration period	End of assigned declaration period
eardata_0.9	eardata version before self-audit	2021-01-01	30/05/2024	2021-01-01	30/05/2024	
eardata_1.0	eardata version with self-audit	2021-05-31	9999-12-31	2021-01-01	31/12/9999	

As can be seen from the above, once eardata_1.0 is live, the eardata_0.9 XSD version can no longer be used, and all declarations and self-audits/corrections must be submitted based on eardata_1.0 XSD, even retroactively.

The check may give the following results:

- If the XSD version specified in the request does not exist, then the manageDeclarationUpload endpoint will not return a declarationUploadId to the user.
- If the XSD version specified in the request exists, but is not valid at the time of submission, i.e. the end of XSD validity < the time of submission, then the manageDeclarationUpload endpoint will not return a declarationUploadId to the user.
- If the XSD version exists and is valid:
 - if the declaration period specified in the request does not fall within the interval assigned to the given XSD, the manageDeclarationUpload endpoint will not return a declarationUploadId to the user.
 - if the declaration period specified in the request falls within the interval assigned to the given XSD, the manageDeclarationUpload endpoint will return a declarationUploadId to the user.

In addition to all these checks, each XSD version has specific validation rules, so the validation rules for the XSD version used to submit the declaration for the given period are run to validate the declaration.

Diagram	
	 <pre> classDiagram class ManageDeclarationUploadReq... class header class user class software class partitionCount class contentHash class declarationSchema class attachmentIdList class xsdVersion class requestPeriodStart class requestPeriodEnd ManageDeclarationUploadReq... "..." --> header : ManageDeclarationUploadReq... "..." --> user : ManageDeclarationUploadReq... "..." --> software : ManageDeclarationUploadReq... "..." --> partitionCount : ManageDeclarationUploadReq... "..." --> contentHash : ManageDeclarationUploadReq... "..." --> declarationSchema : ManageDeclarationUploadReq... "..." --> attachmentIdList : ManageDeclarationUploadReq... "..." --> xsdVersion : ManageDeclarationUploadReq... "..." --> requestPeriodStart : ManageDeclarationUploadReq... "..." --> requestPeriodEnd : </pre>
Namespace	http://schemas.nav.gov.hu/EAR/1.0/api
Type	ManageDeclarationUploadRequestType extension
Features	content complex
Tags	<u>header</u> <u>user</u> <u>software</u> <u>partitionCount</u> <u>contentHash</u> <u>declarationSchema</u> <u>attachmentIdList</u> <u>xsdVersion</u> <u>requestPeriodStart</u> <u>requestPeriodEnd</u>
Content of	Root element of the POST /manageDeclarationUpload REST operation request

element **ManageDeclarationUploadResponse**

The structure of the `/manageDeclarationUpload` operation response is contained within the `ManageDeclarationUploadRequest` element.

In addition to the fields of `BasicResponseType`, it also contains the unique identifier of upload initiation, the start and end of the validity period of the identifier, and, if there was a valid upload identifier at the time the upload request was submitted, the annulled upload identifier.



Diagram	
Namespace	http://schemas.nav.gov.hu/EAR/1.0/api
Type	<u>ManageDeclarationUploadResponseType</u> extension
Features	content complex
Tags	<u>header</u> <u>result</u> <u>declarationUploadId</u> <u>declarationUploadValidFrom</u> <u>declarationUploadValidTo</u> <u>cancelledDeclarationUploadId</u>
Content of	Root element of the POST /manageDeclarationUpload REST operation response

Definition and related requirements

1. In the *partitionCount* tag, the number of partitions planned to upload should be sent. Currently, the maximum value of this field is 16. If a higher value is submitted, the request will be rejected.
2. The *contentHash* is of the complex type *CryptoType*. In addition to the hash value, the type must have the mandatory attribute *cryptoType*, which specifies the hashing algorithm.
 - 3. The accepted hashing algorithm is SHA3-512.
4. In the *declarationSchema* tag, the type of declaration must be specified, which may only have the value *VAT_DECLARATION* currently.
5. The *AttachmentIdList* is of complex type *AttachmentIdListType*, in which the uploaded attachments can be specified with the *claimCheckId* identifier returned after uploading.
6. In the *xsdVersion* tag, the XSD version used to submit the declaration must be specified. If the XSD version was not valid at the time the declaration was submitted, the request will be rejected.
7. In the *requestPeriodStart* tag, the start date of the declaration period must be specified. If the date here does not match the start of the declaration period in the declaration header data, or if it falls outside the declaration period for the relevant XSD version, the request will be rejected.
8. In the *requestPeriodEnd* tag, the end date of the declaration period must be specified. If the date here does not match the end of the declaration period in the declaration header data, or if it falls outside the declaration period for the relevant XSD version, the request will be rejected.
9. The validity of the declarationUploadId received is 3 days (72 hours) from the time of issue.

3.3.3 /manageDeclarationPartition operation

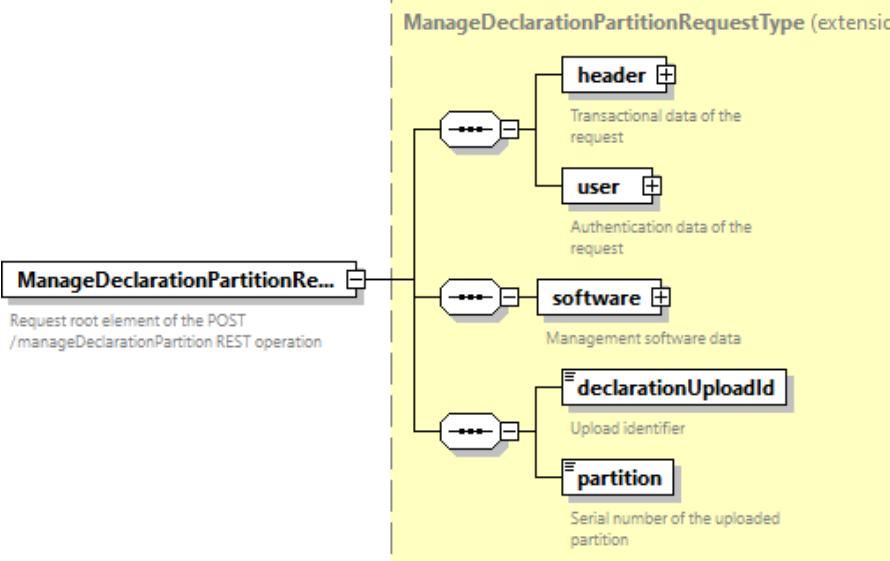
/manageDeclarationPartition is the operation used for submitting the VAT declaration analytics. This can be used to submit the declaration analytics partition to NAV.

No additional permission other than authentication is required to upload the attachment, since requesting *declarationUploadId* can already be done via a permission-protected endpoint.

element **ManageDeclarationPartitionRequest**

The structure of the */manageDeclarationPartition* operation is contained within the **ManageDeclarationPartitionRequest** element.

This type extends *BasicEVatRequestType*, meaning that in addition to the elements it contains, it will also incorporate a the upload ID and the serial number of the partition being uploaded.

<i>Diagram</i>	 <pre> classDiagram class ManageDeclarationPartitionRequestType { <<extension of BasicEVatRequestType>> header user software declarationUploadId partition } class ManageDeclarationPartitionRequestType { <<Request root element of the POST /manageDeclarationPartition REST operation>> } ManageDeclarationPartitionRequestType < -- ManageDeclarationPartitionRequestType </pre>
<i>Namespace</i>	http://schemas.nav.gov.hu/EAR/1.0/api
<i>Type</i>	<u>ManageDeclarationPartitionRequestType</u> extension
<i>Features</i>	content complex
<i>Tags</i>	header user software declarationUploadId partition
<i>Content of</i>	Root element of the POST /manageDeclarationPartition REST operation request

element **ManageDeclarationPartitionResponse**

The structure of the */manageDeclarationPartition* operation response is contained within the **ManageDeclarationPartitionResponse** element.

The type extends *BasicResponseType*. In addition to the fields of *BasicResponseType*, it contains the unique identifier of upload initiation, and the serial number of the partition uploaded.



Diagram	
Namespace	http://schemas.nav.gov.hu/EAR/1.0/api
Type	<u>ManageDeclarationPartitionResponseType</u> extension
Features	content complex
Tags	<u>header</u> <u>result</u> <u>declarationUploadId</u> <u>partition</u>
Content of	Root element of the POST /manageDeclarationPartition REST operation response

Definition and related requirements

1. The *declarationUploadId* tag must contain the identifier received in the */manageDeclarationUpload* response. Naturally, the validity period should be taken into account. If the ID is expired or does not exist, the upload request will be rejected.
2. The *partition* tag indicates the serial number of the partition to be uploaded. The serial number of the partition must not exceed the value of the *partitionCount* field specified in the */manageDeclarationUpload* request. For example, if 5 partitions were declared previously in the */manageDeclarationUpload* request, this field can have a value from 1 to 5.
3. Only one partition per serial number can be uploaded.
4. The size of a partition should not exceed 128MB; if this does occur, the connection will be terminated from the server side.
5. The POST */manageDeclarationUpload* endpoint should be sent a request with the format '*multipart/form-data*' (RFC 2388), consisting of two messages. The first one is of type '*application/xml*'; this is the request defined by the *ManageDeclarationPartitionRequest* schema element. The type of the other one is '*application/octet-stream*'; this contains the attachment.
6. Partitions are generated as follows. The client calculates the SHA3-512 hash value of the declaration XML generated based on the finalized *earData.xsd*, and compresses the object into gzip format. The client then determines how many partitions the declaration to be uploaded will consist of, depending on whether the size of the compressed object exceeds the maximum partition size. The client splits the compressed file into byte arrays, respecting the maximum partition size, and uploads it as a bytestream in the '*application/octet-stream*' message of the multipart REST request.
7. To successfully finalise VAT analytics (*/manageDeclarationFinalize*), all partitions must be received in full.

3.3.4 /manageDeclarationFinalize operation

/manageDeclarationFinalize is the operation used to finalise the submission of VAT analytics. This is used to indicate that client-side partition uploads are complete and processing may begin.

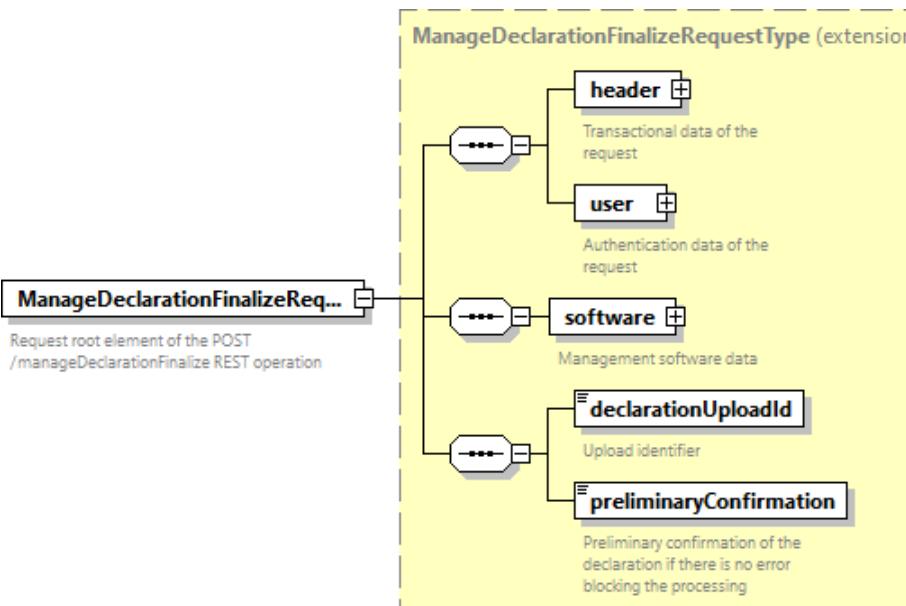
No additional permission other than authentication is required to upload the attachment.

Required permission: VM2M Declaration management

element ManageDeclarationFinalizeRequest

The structure of the */manageDeclarationFinalize* operation is contained within the *ManageDeclarationFinalizeRequest* element.

This type extends *BasicEVatRequestType*, meaning that in addition to the elements it contains, it will also incorporate the upload ID and a flag indicating prior approval.

<p><i>Diagram</i></p> 	
<i>Namespace</i>	http://schemas.nav.gov.hu/EAR/1.0/api
<i>Type</i>	<u>ManageDeclarationFinalizeRequestType</u> extension
<i>Features</i>	content complex
<i>Tags</i>	<u>header</u> <u>user</u> <u>software</u> <u>declarationUploadId</u> <u>preliminaryConfirmation</u>
<i>Content of</i>	Root element of the POST <i>/manageDeclarationFinalize</i> REST operation request

element ManageDeclarationFinalizeResponse

The structure of the */manageDeclarationFinalize* operation response is contained within the *ManageDeclarationFinalizeResponse* element.

The type extends *BasicResponseType*. In addition to the fields of *BasicResponseType*, the type contains the unique identifier of the processing in case of a successful request.

<i>Diagram</i>	<pre> classDiagram class ManageDeclarationFinalizeRes... class ManageDeclarationFinalizeResponseType { <<extension>> header result declarationProcessingId } ManageDeclarationFinalizeRes... "1" -- "1" ManageDeclarationFinalizeResponseType </pre>
<i>Namespace</i>	http://schemas.nav.gov.hu/EAR/1.0/api
<i>Type</i>	<u>ManageDeclarationFinalizeResponseType</u> extension
<i>Features</i>	content complex
<i>Tags</i>	<u>header</u> <u>result</u> <u>declarationProcessingId</u>
<i>Content of</i>	Root element of the POST /manageDeclarationFinalize REST operation response

Definition and related requirements

1. The *declarationUploadId* tag must contain the identifier received in the */manageDeclarationUpload* response. Naturally, the validity period should be taken into account. If the ID is expired or does not exist, the upload request will be rejected.
2. When the operation is called, the system checks whether all the partitions and attachments previously declared for the upload ID have been received. If these are all present, processing begins. In all other cases, the system will return an error.
3. The value of the *preliminaryConfirmation* tag is required to be *false* (reserved for future use).
4. If the call is successful, the declaration is saved with the *RECEIVED* status. The status of the declaration can be checked by calling the */queryDeclarationProcessingStatus* operation using the *declarationProcessingId* value.

3.3.5 /manageDeclarationSubmission operation

The */manageDeclarationSubmission* operation is used for submitting a VAT declaration.

element **ManageDeclarationSubmissionRequest**

The structure of the */manageDeclarationSubmission* operation is contained within the *ManageDeclarationSubmissionRequest* element.

This type extends *BasicEVatRequestType*, meaning that in addition to the elements it contains, it will also incorporate the processing ID.

Required permission: VM2M Declaration approval



<i>Diagram</i>	<p>ManageDeclarationSubmissionRequestType (extension)</p> <p>header + Transactional data of the request</p> <p>user + Authentication data of the request</p> <p>software + Management software data</p> <p>declarationProcessingId Processing identifier</p> <p>Request root element of the POST /manageDeclarationSubmission REST operation</p>
<i>Namespace</i>	http://schemas.nav.gov.hu/EAR/1.0/api
<i>Type</i>	<u>ManageDeclarationSubmissionRequestType</u> extension
<i>Features</i>	content complex
<i>Tags</i>	<u>header</u> <u>user</u> <u>software</u> <u>declarationProcessingId</u> <u>recipientTaxIdentificationNumber</u>
<i>Content of</i>	Root element of the POST /manageDeclarationSubmission REST operation request

element **ManageDeclarationSubmissionResponse**

The structure of the /manageDeclarationSubmission operation response is contained within the ManageDeclarationSubmissionResponseType element.

The type extends *BasicResponseType*. In addition to the fields of *BasicResponseType*, the type contains the hash of the uploaded file, the summary data of the declaration, and the declaration data compiled based on the VAT analytics in BEVFELD XML format in case of a successful request.

<i>Diagram</i>	<p>ManageDeclarationSubmissionResponseType (extension)</p> <p>header + Transactional data of the response</p> <p>result + Basic result data</p> <p>contentHash + Name and value of the hash algorithm calculated for the uploaded content</p> <p>declarationSummary + Summary data of the declaration</p> <p>bevfeldData BEVFELD XML data based on VAT analytics</p> <p>Response root element of the POST /manageDeclarationSubmission REST operation</p>
----------------	---

<i>Namespace</i>	http://schemas.nav.gov.hu/EAR/1.0/api
<i>Type</i>	<u>ManageDeclarationSubmissionResponseType</u> extension
<i>Features</i>	<i>content</i> complex
<i>Tags</i>	header result contentHash declarationSummary bevfeldData
<i>Content of</i>	Root element of the POST /manageDeclarationSubmission REST operation response

Definition and related requirements

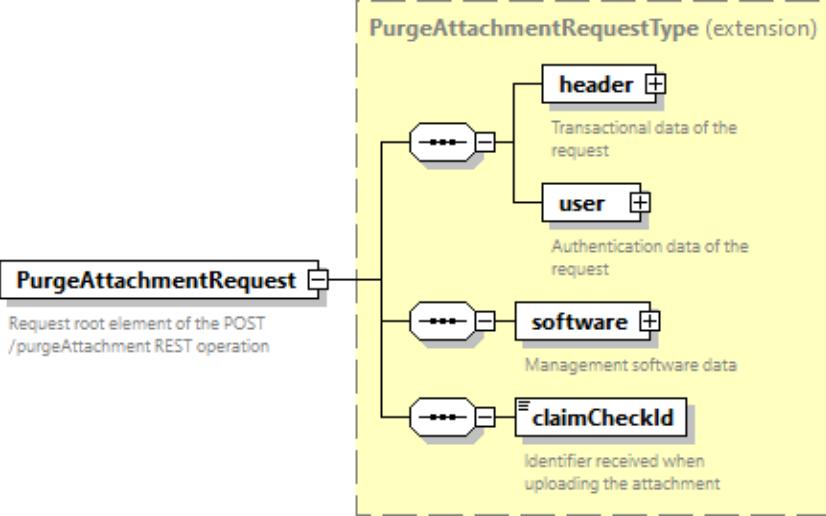
1. The *declarationProcessingId* must be filled with the processing identifier (*declarationProcessingId*) returned in the response to the */manageDeclarationFinalize* operation.
2. Only declarations with '*FINISHED*' status can be submitted.
3. The declaration will be in '*FINISHED*' status for 30 days. If it is not submitted within this period, it will automatically change to '*ABORTED*' status. After this, it is no longer possible to submit the declaration, and the process must be restarted.

3.3.6 /purgeAttachment operation

The /purgeAttachment operation can be used to delete previously uploaded declaration attachments.

No additional permission other than authentication is required to delete the attachment.

element PurgeAttachmentRequest

<i>Diagram</i>	 <pre> classDiagram class PurgeAttachmentRequest { <<Request root element of the POST /purgeAttachment REST operation>> } extensionType PurgeAttachmentRequestType { <<PurgeAttachmentRequestType (extension)>> attribute header attribute user attribute software attribute claimCheckId } PurgeAttachmentRequest "1" -- "1" PurgeAttachmentRequestType </pre>
<i>Namespace</i>	http://schemas.nav.gov.hu/EAR/1.0/api
<i>Type</i>	<u>PurgeAttachmentRequestType</u> extension
<i>Features</i>	<i>content</i> complex
<i>Tags</i>	header user software claimCheckId
<i>Content of</i>	Root element of the POST /purgeAttachment REST operation request



element **PurgeAttachmentResponse**

<i>Diagram</i>	<p>The diagram illustrates the structure of the PurgeAttachmentResponseType. It is an extension of the base PurgeAttachmentResponse element. The structure includes a header component (Transactional data of the response) and a result component (Basic result data).</p>
<i>Namespace</i>	http://schemas.nav.gov.hu/EAR/1.0/api
<i>Type</i>	<u>PurgeAttachmentResponseType</u> extension
<i>Features</i>	content complex
<i>Tags</i>	<u>header</u> <u>result</u>
<i>Content of</i>	Root element of the POST /purgeAttachment REST operation response

Definition and related requirements

1. The claimCheckId provided can be used to release the quota available to the taxpayer.
2. Attachments associated to a VAT analysis being processed cannot be deleted.

3.3.7 /queryAttachmentList operation

Using the queryAttachmentList operation, the taxpayer can list the attachments in their repository.

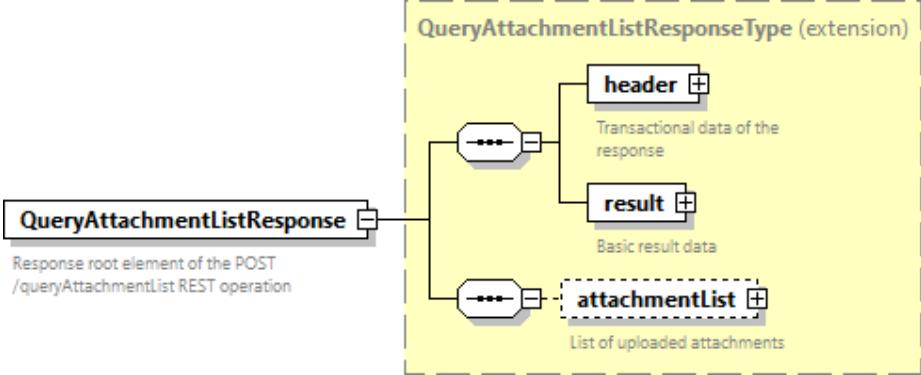
No additional permission other than authentication is required to list attachments.

element **QueryAttachmentListRequest**

<i>Diagram</i>	<p>The diagram illustrates the structure of the QueryAttachmentListRequestType. It is an extension of the base QueryAttachmentListRequest element. The structure includes a header component (Transactional data of the request), a user component (Authentication data of the request), and a software component (Management software data).</p>
<i>Namespace</i>	http://schemas.nav.gov.hu/EAR/1.0/api

Type	<u>QueryAttachmentListRequestType</u> extension
Features	content complex
Tags	<u>header</u> <u>user</u> <u>software</u>
Content of	Root element of the POST /queryAttachmentList REST operation request

element **QueryAttachmentListResponse**

<i>Diagram</i>	 <pre> classDiagram class QueryAttachmentListResponseType { header result attachmentList } class QueryAttachmentListResponse { <<Response root element of the POST /queryAttachmentList REST operation>> } QueryAttachmentListResponse "1..*" --> "1..*" QueryAttachmentListResponseType : header result attachmentList </pre>
Namespace	http://schemas.nav.gov.hu/EAR/1.0/api
Type	<u>QueryAttachmentListResponseType</u> extension
Features	content complex
Tags	<u>header</u> <u>result</u> <u>attachmentList</u>
Content of	Root element of the POST /queryAttachmentList REST operation response

Definition and related requirements

1. If no records can be returned based on the search parameters, it returns an empty response.

3.3.8 /queryCustomsDeclarationDigest operation

/queryCustomsDeclarationDigest is the operation for querying customs decisions (CDSPS digest).

Required permission: VM2M Convenience services, queries

element **QueryCustomsDeclarationDigestRequest**

The structure of the /queryCustomsDeclarationDigest operation is contained within the QueryCustomsDeclarationDigestRequest element.

This type extends *BasicEVatRequestType*, meaning that in addition to the elements it contains, it will also incorporate the numbers of pages to query and the query parameters.



<i>Diagram</i>	<p>Request root element of the POST /queryCustomsDeclarationDigest REST operation</p>
<i>Namespace</i>	http://schemas.nav.gov.hu/EAR/1.0/api
<i>Type</i>	<u>QueryCustomsDeclarationDigestRequestType</u> extension
<i>Features</i>	<i>content</i> complex
<i>Tags</i>	header user software page declarationDirection declarationqueryParams
<i>Content of</i>	Root element of the POST /queryCustomsDeclarationDigest REST operation request

element **QueryCustomsDeclarationDigestResponse**

<i>Diagram</i>	<p>Response root element of the POST /queryCustomsDeclarationDigest REST operation</p>
<i>Namespace</i>	http://schemas.nav.gov.hu/EAR/1.0/api
<i>Type</i>	<u>QueryCustomsDeclarationDigestResponseType</u> extension
<i>Features</i>	<i>content</i> complex

<i>Tags</i>	header result availablePage currentPage declarationDigest
<i>Content of</i>	Root element of the POST /queryCustomsDeclarationDigest REST operation response

Definition and related requirements

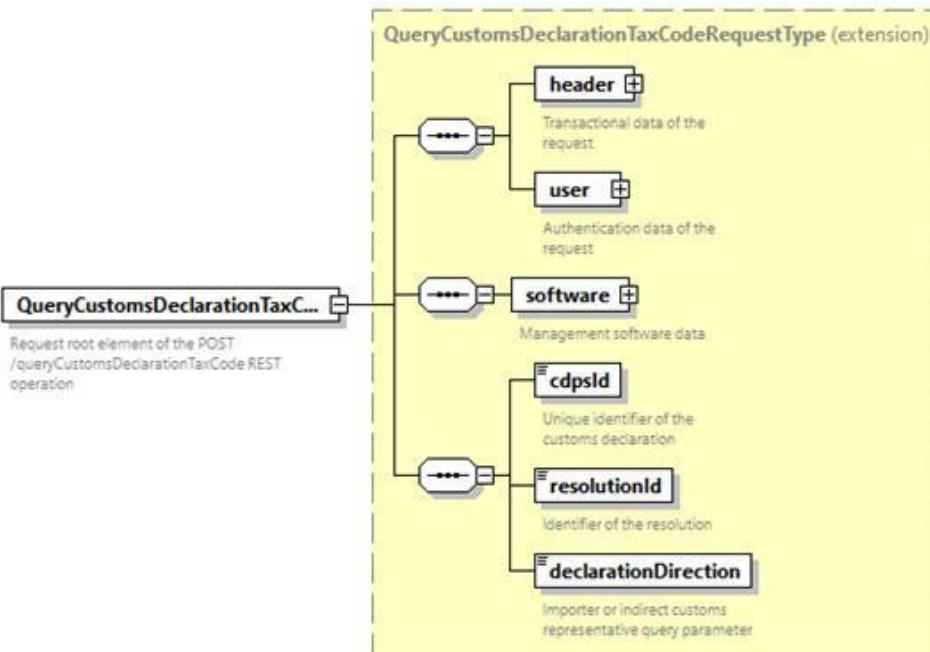
1. In the mandatoryDeclarationQueryParams node, the received declarationDateFrom <= declarationDateTo.
2. The maximum difference allowed between declarationDateFrom and declarationDateTo is 35 days.
3. The page size used is 100.
4. If no records can be returned based on the search parameters, or there is no data on the requested page number, the service returns an empty list.
5. The request will only be served if the CDPS decision has already been processed (catalogued, tax coded) electronically. Cataloguing, tax coding is typically done within 24 hours of the decision being issued

3.3.9 /queryCustomsDeclarationTaxCode operation

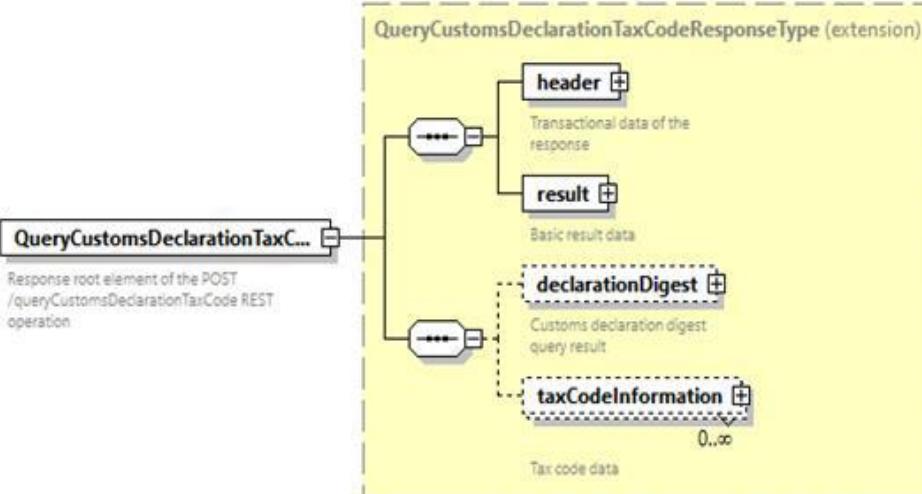
The /queryCustomsDeclarationTaxCode operation is used to query the tax codes for a CDPS decision.

Required permission: VM2M Convenience services, queries

element QueryCustomsDeclarationTaxCodeRequest

<i>Diagram</i>	 <pre> classDiagram class QueryCustomsDeclarationTaxCodeRequestType { header user software cdpsId resolutionId declarationDirection } class QueryCustomsDeclarationTaxCodeRequestType { <> QueryCustomsDeclarationTaxCodeRequestType } QueryCustomsDeclarationTaxCodeRequestType < -- QueryCustomsDeclarationTaxCodeRequestType </pre>
<i>Namespace</i>	http://schemas.nav.gov.hu/EAR/1.0/api
<i>Type</i>	QueryCustomsDeclarationTaxCodeRequestType extension
<i>Features</i>	<i>content</i> complex
<i>Tags</i>	header user software cdpsId resolutionId declarationDirection
<i>Content of</i>	Root element of the POST /queryCustomsDeclarationTaxCode REST operation request

element QueryCustomsDeclarationTaxCodeResponse

<i>Diagram</i>	 <pre> classDiagram class QueryCustomsDeclarationTaxCodeResponseType { header +Transactional data of the response result +Basic result data declarationDigest +Customs declaration digest query result taxCodeInformation 0..∞ Tax code data } QueryCustomsDeclarationTaxCodeResponseType < -- QueryCustomsDeclarationTaxC... </pre>
<i>Namespace</i>	http://schemas.nav.gov.hu/EAR/1.0/api
<i>Type</i>	QueryCustomsDeclarationTaxCodeResponseType extension
<i>Feature</i>	<i>content</i> complex
<i>Tags</i>	header result declarationDigest taxCodeInformation
<i>Content of</i>	Root element of the POST /queryCustomsDeclarationTaxCode REST operation response

Definition and related requirements

1. If no records are returned according to the search parameters, the service returns an empty list.
2. The request will only be served if the CDPS decision has already been processed (catalogued) electronically.

3.3.10 /queryDeclarationData operation

The /queryDeclarationData operation is used to download declaration data.

Required permission: VM2M Convenience services, queries



element **QueryDeclarationDataRequest**

<i>Diagram</i>	<p>QueryDeclarationDataRequest (extension)</p> <p>header + Transactional data of the request</p> <p>user + Authentication data of the request</p> <p>software + Management software data</p> <p>declarationProcessingId Processing identifier</p>
<i>Namespace</i>	http://schemas.nav.gov.hu/EAR/1.0/api
<i>Type</i>	QueryDeclarationDataRequestType extension
<i>Features</i>	<i>content</i> complex
<i>Tags</i>	header user software declarationProcessingId
<i>Content of</i>	Root element of the POST /queryDeclarationData REST operation request

element **QueryDeclarationDataResponse**

<i>Diagram</i>	<p>QueryDeclarationDataResponseType (extension)</p> <p>header + Transactional data of the response</p> <p>result + Basic result data</p> <p>declarationData + Declaration data</p>
<i>Namespace</i>	http://schemas.nav.gov.hu/EAR/1.0/api
<i>Type</i>	QueryDeclarationDataResponseType extension
<i>Features</i>	<i>content</i> complex
<i>Tags</i>	header result declarationData
<i>Content of</i>	Root element of the POST /queryDeclarationData REST operation response

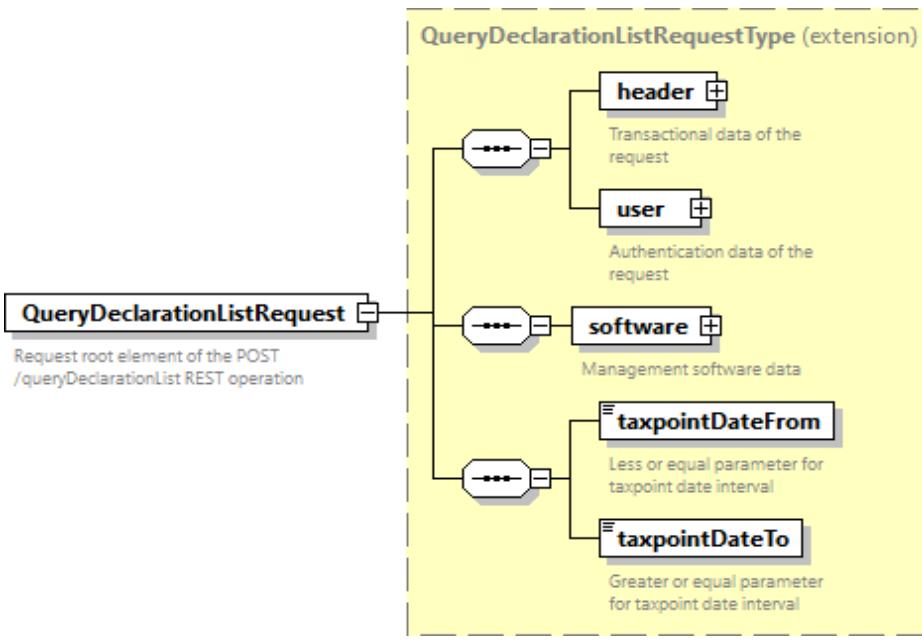
Definition and related requirements

- If no records can be returned based on the search parameters, the service returns an empty list.

3.3.11 /queryDeclarationList operation

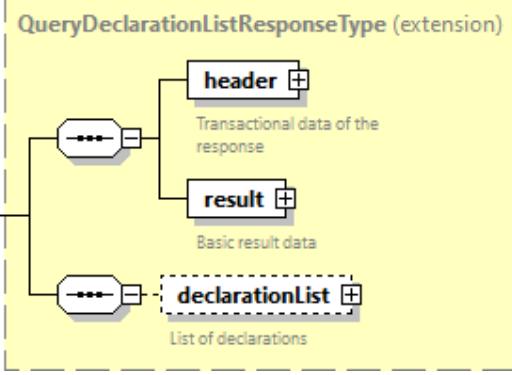
Required permission: VM2M Convenience services, queries

element **QueryDeclarationListRequest**

<p><i>Diagram</i></p>	 <pre> classDiagram class QueryDeclarationListRequest { <<Request root element of the POST /queryDeclarationList REST operation>> } class QueryDeclarationListRequestType { <<extension>> } class header { <<Transactional data of the request>> } class user { <<Authentication data of the request>> } class software { <<Management software data>> } class taxpointDateFrom { <<Less or equal parameter for taxpoint date interval>> } class taxpointDateTo { <<Greater or equal parameter for taxpoint date interval>> } QueryDeclarationListRequest "1" --> "1" QueryDeclarationListRequestType QueryDeclarationListRequestType "1" --> "1" header QueryDeclarationListRequestType "1" --> "1" user QueryDeclarationListRequestType "1" --> "1" software QueryDeclarationListRequestType "1" --> "1" taxpointDateFrom QueryDeclarationListRequestType "1" --> "1" taxpointDateTo </pre>
<i>Namespace</i>	http://schemas.nav.gov.hu/EAR/1.0/api
<i>Type</i>	<u>QueryDeclarationListRequestType</u> extension
<i>Features</i>	<i>content</i> complex
<i>Tags</i>	header user software taxpointDateFrom taxpointDateTo
<i>Content of</i>	Root element of the POST /queryDeclarationList REST operation request



element **QueryDeclarationListResponse**

Diagram	
	<p>QueryDeclarationListResponse Response root element of the POST /queryDeclarationList REST operation</p>
<i>Namespace</i>	http://schemas.nav.gov.hu/EAR/1.0/api
<i>Type</i>	QueryDeclarationListResponseType extension
<i>Features</i>	content complex
<i>Tags</i>	header result declarationList
<i>Content of</i>	Root element of the POST /queryDeclarationList REST operation response

Definition and related requirements

1. In the request, the received taxpointDateFrom <= taxpointDateTo.
2. The maximum difference allowed between taxpointDateFrom and taxpointDateTo is 35 days.
3. taxpointDateFrom cannot be in the future.
4. The maximum number of declarations is 100.
5. If no records are returned according to the search parameters, the service returns an empty list.

3.3.12 /queryDeclarationProcessingStatus operation

The /queryDeclarationProcessingStatus operation is used to query the processing result of a declaration.

Required permission: VM2M Convenience services, queries

element **QueryDeclarationProcessingStatusRequest**

The structure of the /queryDeclarationProcessingStatus operation request is contained within the QueryDeclarationProcessingStatusRequest element.

This type extends *BasicEVatRequestType*, meaning that in addition to the elements it contains, it will also incorporate a processing ID.



<i>Diagram</i>	<p>The diagram illustrates the structure of the <code>QueryDeclarationProcessingStatusRequestType</code> extension. It features a root element <code>QueryDeclarationProcessingSt...</code> which contains four components: <code>header</code>, <code>user</code>, <code>software</code>, and <code>declarationProcessingId</code>. Each component is described with its purpose: <code>header</code> for transactional data, <code>user</code> for authentication data, <code>software</code> for management software data, and <code>declarationProcessingId</code> for processing identifiers.</p>
<i>Namespace</i>	http://schemas.nav.gov.hu/EAR/1.0/api
<i>Type</i>	<code>QueryDeclarationProcessingStatusRequestType</code> extension
<i>Features</i>	content complex
<i>Tags</i>	<code>header</code> <code>user</code> <code>software</code> <code>declarationProcessingId</code>
<i>Content of</i>	Root element of the POST /queryDeclarationProcessingStatus REST operation request

element `QueryDeclarationProcessingStatusResponse`

The structure of the `/queryDeclarationProcessingStatus` operation response is contained within the `QueryDeclarationProcessingStatusResponseType` element.

The type extends `BasicResponseType`. In case of a successful request – provided there is a result – the type contains the data and processing status of the declaration in addition to the fields of `BasicResponseType`.

<i>Diagram</i>	<p>The diagram illustrates the structure of the <code>QueryDeclarationProcessingStatusResponseType</code> extension. It features a root element <code>QueryDeclarationProcessingSt...</code> which contains three components: <code>header</code>, <code>result</code>, and <code>declarationProcessingStatus</code>. <code>header</code> contains transactional data of the response. <code>result</code> contains basic result data. <code>declarationProcessingStatus</code> contains declaration processing status.</p>
<i>Namespace</i>	http://schemas.nav.gov.hu/EAR/1.0/api
<i>Type</i>	<code>QueryDeclarationProcessingStatusResponseType</code> extension

<i>Features</i>	<i>content</i> complex
<i>Tags</i>	header result declarationProcessingStatus
<i>Content of</i>	Root element of the POST /queryDeclarationProcessingStatus REST operation response

Definition and related requirements

1. When providing the identifier received in the response of the */manageDeclarationFinalize* operation in the *declarationProcessingId* tag, the system returns the current processing status of the declaration. If *declarationProcessingId* is incorrect, the service returns an empty response.
2. The values of the *declarationStatus* field have the following meanings:
3. RECEIVED = The uploaded declaration is awaiting processing.
4. PROCESSING = The uploaded declaration is being checked.
5. BEVFELD_CHECK = The uploaded declaration is being checked by NAV.
6. FINISHED = The uploaded declaration has been successfully checked.
7. SUBMITTED = The declaration has been successfully submitted.
8. ABORTED = There is an error in the uploaded declaration that prevents its submission.
9. The *longRunWarningStatus* field indicates the status of long run checks. The completion of these checks is not required for submitting the declaration, but their results can be a useful source of information regarding the correctness of the declaration. The *longRunStatusCode* field can have the following values:
 10. POST_EVALUATOR_PROCESSING = The processing of long run checks on items in the declaration has begun.
 11. POST_EVALUATED = The long run checks on the items in the declaration have been successfully completed.
 12. POST_EVALUATOR_FAILED = A technical error occurred while checking the items in the declaration, and the check failed as a result.
 13. POST_EVALUATOR_ABORTED = There is an error in the declaration that prevents its submission, causing the checking of the items to be aborted.

3.3.13 /queryDocumentList operation

The /queryDocumentList operation starts an asynchronous process, as a result of which the system queries the taxpayer's vouchers for the declaration period.

Required permission: VM2M Convenience services, queries



element **QueryDocumentListRequest**

<i>Diagram</i>	<pre>classDiagram class QueryDocumentListRequest { <<Request root element of the POST /queryDocumentList REST operation>> } class QueryDocumentListRequestType { <<extension>> header user software taxpointDateFrom taxpointDateTo } QueryDocumentListRequest "1" -- "*" QueryDocumentListRequestType QueryDocumentListRequestType "1" -- "*" header QueryDocumentListRequestType "1" -- "*" user QueryDocumentListRequestType "1" -- "*" software QueryDocumentListRequestType "1" -- "*" taxpointDateFrom QueryDocumentListRequestType "1" -- "*" taxpointDateTo</pre>
<i>Namespace</i>	http://schemas.nav.gov.hu/EAR/1.0/api
<i>Type</i>	<u>QueryDocumentListRequestType</u> extension
<i>Features</i>	content complex
<i>Tags</i>	<u>header</u> <u>user</u> <u>software</u> <u>taxpointDateFrom</u> <u>taxpointDateTo</u>
<i>Content of</i>	Root element of the POST /queryDocumentList REST operation request

element **QueryDocumentListResponse**

<i>Diagram</i>	<pre>classDiagram class QueryDocumentListResponse { <<Response root element of the POST /queryDocumentList REST operation>> } class QueryDocumentListResponseType { <<extension>> header result queryId } QueryDocumentListResponse "1" -- "*" QueryDocumentListResponseType QueryDocumentListResponseType "1" -- "*" header QueryDocumentListResponseType "1" -- "*" result QueryDocumentListResponseType "1" -- "*" queryId</pre>
<i>Namespace</i>	http://schemas.nav.gov.hu/EAR/1.0/api
<i>Type</i>	<u>QueryDocumentListResponseType</u> extension
<i>Features</i>	content complex

<i>Tags</i>	<u>header</u> <u>result</u> <u>queryId</u>
<i>Content of</i>	Root element of the POST /queryDocumentList REST operation response

Definition and related requirements

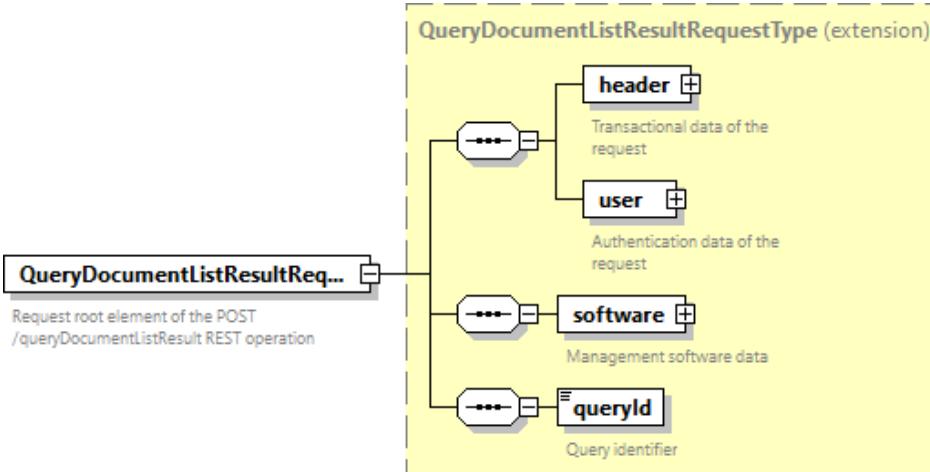
1. In the request, the received taxpointDateFrom <= taxpointDateTo.
2. The maximum difference allowed between taxpointDateFrom and taxpointDateTo is 35 days.
3. If all checks were successful, a *queryId* is returned, which should be used in the /queryDocumentListResult request.

3.3.14 /queryDocumentListResult operation

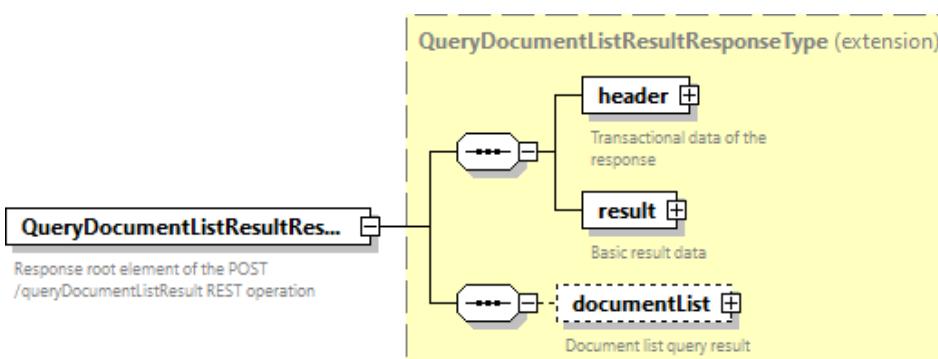
The /queryDocumentListResult operation can be used to query the status and result of an asynchronous process started with the /queryDocumentList operation. As a result of this process, the system will query the taxpayer's vouchers for the period specified in the request.

Required permission: VM2M Convenience services, queries

element **QueryDocumentListResultRequest**

<i>Diagram</i>	 <pre> classDiagram class QueryDocumentListResultReq... QueryDocumentListResultReq... "..." --> header : ... QueryDocumentListResultReq... "..." --> user : ... QueryDocumentListResultReq... "..." --> software : ... } class QueryDocumentListResultRequestType { header user software queryId } Note over QueryDocumentListResultReq...: Request root element of the POST /queryDocumentListResult REST operation </pre>
<i>Namespace</i>	http://schemas.nav.gov.hu/EAR/1.0/api
<i>Type</i>	QueryDocumentListResultRequestType extension
<i>Features</i>	<i>content</i> complex
<i>Tags</i>	<u>header</u> <u>user</u> <u>software</u> <u>queryId</u>
<i>Content of</i>	Root element of the POST /queryDocumentListResult REST operation request

element QueryDocumentListResultResponse

<p><i>Diagram</i></p>  <p>QueryDocumentListResultRes...</p> <p>Response root element of the POST /queryDocumentListResult REST operation</p>	<p>QueryDocumentListResultResponseType (extension)</p> <ul style="list-style-type: none"> header + Transactional data of the response result + Basic result data documentList + Document list query result
<i>Namespace</i>	http://schemas.nav.gov.hu/EAR/1.0/api
<i>Type</i>	QueryDocumentListResultResponseType extension
<i>Features</i>	<i>content</i> complex
<i>Tags</i>	header result documentList
<i>Content of</i>	Root element of the POST /queryDocumentListResult REST operation response

Definition and related requirements

1. The status of the query can be *PROCESSING* if the compilation of data sources is not yet complete, or *DONE*
2. The query results are stored for 24 hours. This is the time available to the taxpayer to query the data.
3. If no records can be returned based on the search parameters, the system returns an empty response.

3.3.15 /queryInvoiceTaxCode operation

The /queryInvoiceTaxCode operation is used to query invoice tax codes and VAT-relevant WARNs.

Required permission: VM2M Convenience services, queries



element **QueryInvoiceTaxCodeRequest**

<i>Diagram</i>	
<i>Namespace</i>	http://schemas.nav.gov.hu/EAR/1.0/api
<i>Type</i>	QueryInvoiceTaxCodeRequestType extension
<i>Features</i>	content complex
<i>Tags</i>	header user software invoiceNumberQuery
<i>Content of</i>	Root element of the POST /queryInvoiceTaxCode REST operation request

element **QueryInvoiceTaxCodeResponse**

<i>Diagram</i>	
----------------	--

<i>Namespace</i>	http://schemas.nav.gov.hu/EAR/1.0/api
<i>Type</i>	QueryInvoiceTaxCodeResponseType extension
<i>Features</i>	<i>content</i> complex
<i>Tags</i>	header result invoiceDigest taxCodeInformation vatRelevantWarning
<i>Content of</i>	Root element of the POST /queryInvoiceTaxCode REST operation response

Definition and related requirements

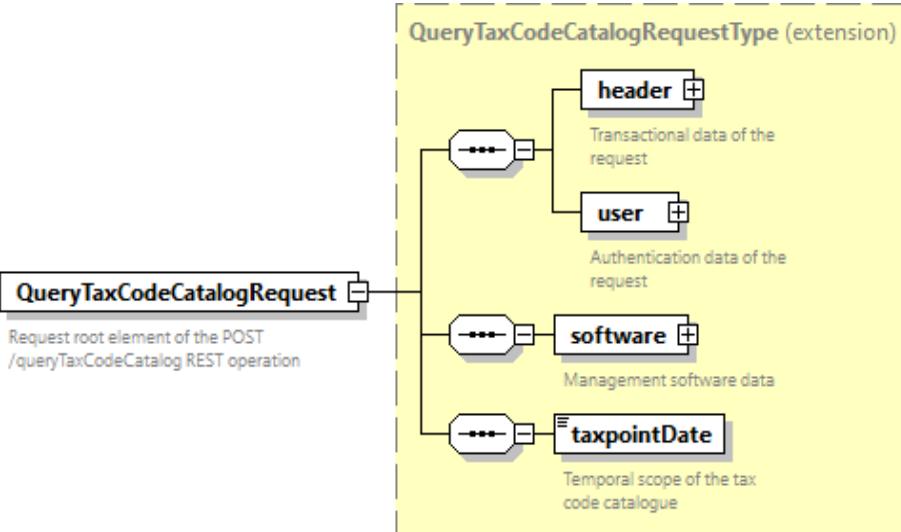
1. If no invoice can be returned based on the search parameters, an empty response is returned.
2. If based on the search parameters, the invoice exists, but there are no tax codes or WARNs associated with it, then the response will not contain taxCodeInformation or vatRelevantWarning nodes.
3. The SupplierTaxNumber field can only be filled if invoiceDirection = INBOUND. Otherwise, the system will return an error.

3.3.16 /queryTaxCodeCatalog operation

The queryTaxCodeCatalog operation can be used to retrieve the tax code catalogue in force on a given day.

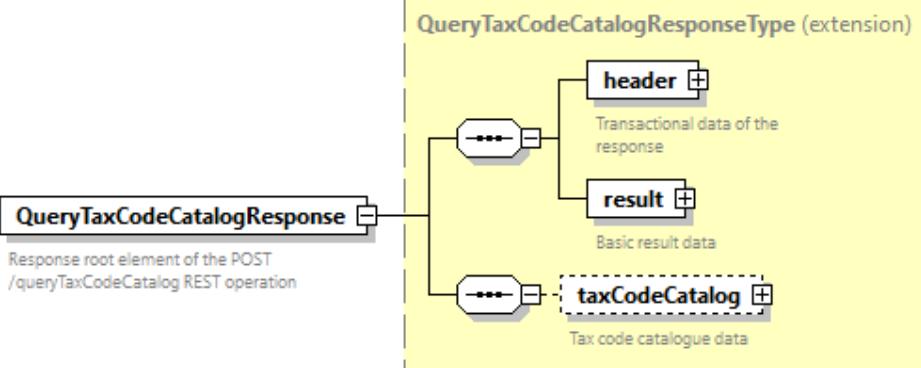
Required permission: VM2M Convenience services, queries

element **QueryTaxCodeCatalogRequest**

<i>Diagram</i>	 <pre> classDiagram class QueryTaxCodeCatalogRequestType { <<extension>> header user software taxpointDate } class QueryTaxCodeCatalogRequest { <<Request root element of the POST /queryTaxCodeCatalog REST operation>> } QueryTaxCodeCatalogRequest --> QueryTaxCodeCatalogRequestType </pre>
<i>Namespace</i>	http://schemas.nav.gov.hu/EAR/1.0/api
<i>Type</i>	QueryTaxCodeCatalogRequestType extension
<i>Features</i>	<i>content</i> complex
<i>Tags</i>	header user software taxpointDate
<i>Content of</i>	Root element of the POST /queryTaxCodeCatalog REST operation request



element **QueryTaxCodeCatalogResponse**

<i>Diagram</i>	
<i>Namespace</i>	http://schemas.nav.gov.hu/EAR/1.0/api
<i>Type</i>	QueryTaxCodeCatalogResponseType extension
<i>Features</i>	content complex
<i>Tags</i>	header result taxCodeCatalog
<i>Content of</i>	Root element of the POST /queryTaxCodeCatalog REST operation response

Definition and related requirements

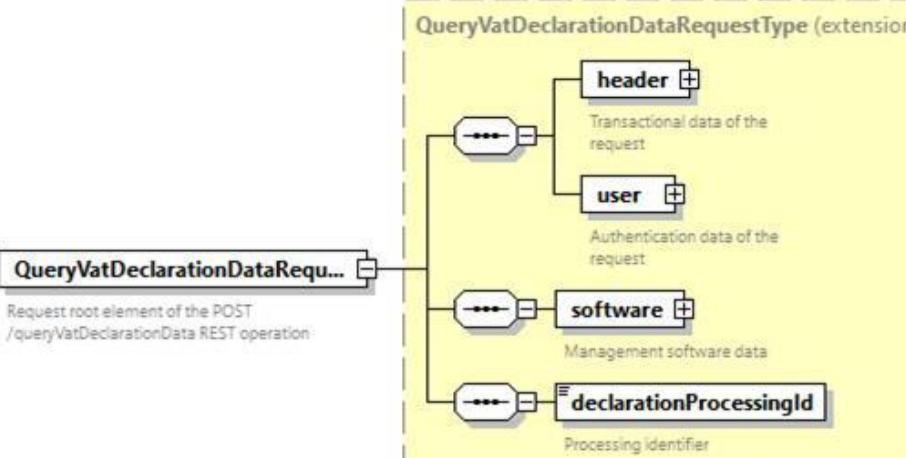
1. If no valid tax code catalogue for the taxpointDate specified can be returned based on the search parameters, an empty response is returned.

3.3.17 /queryVatDeclarationData operation

The /queryVatDeclarationData operation is used to download the declaration analytics that have already been submitted.

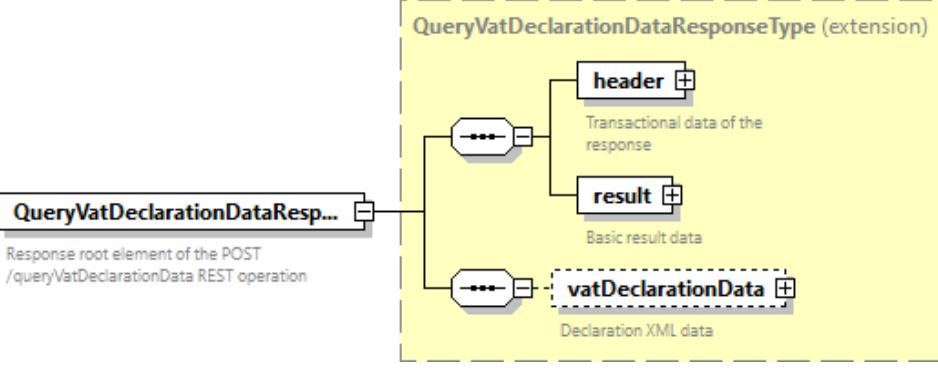
Required permission: VM2M Convenience services, queries

element **QueryVatDeclarationDataRequest**

<i>Diagram</i>	
	QueryVatDeclarationDataReq... Request root element of the POST /queryVatDeclarationData REST operation

<i>Namespace</i>	http://schemas.nav.gov.hu/EAR/1.0/api
<i>Type</i>	QueryVatDeclarationDataRequestType extension
<i>Features</i>	<i>content</i> complex
<i>Tags</i>	header user software declarationProcessingId
<i>Content of</i>	Root element of the POST /queryVatDeclarationData REST operation request

element **QueryVatDeclarationDataResponse**

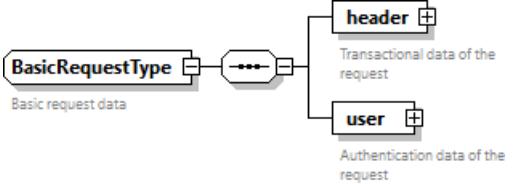
<i>Diagram</i>	
<i>Namespace</i>	http://schemas.nav.gov.hu/EAR/1.0/api
<i>Type</i>	QueryVatDeclarationDataResponseType extension
<i>Features</i>	<i>content</i> complex
<i>Tags</i>	header result vatDeclarationData
<i>Content of</i>	Root element of the POST /queryVatDeclarationData REST operation response

Definition and related requirements

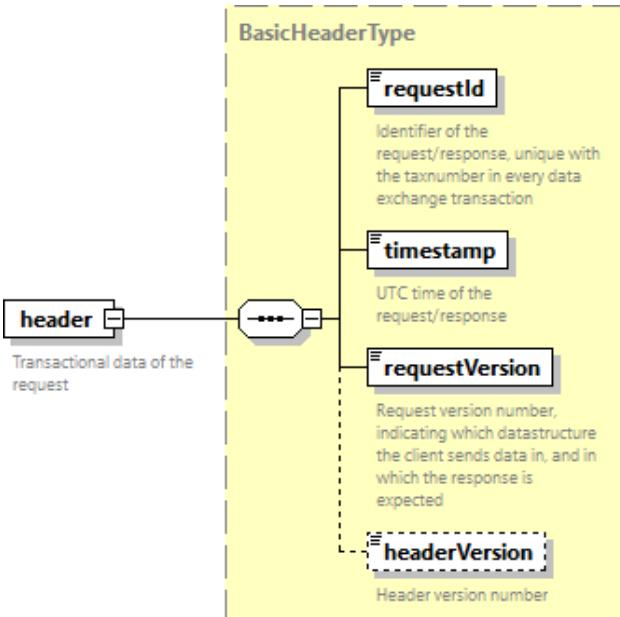
1. If no analytics can be returned based on the search parameter, an empty response is returned.
2. The query is supported for processing IDs (*declarationProcessingId*) with status 'SUBMITTED'.
3. Querying is possible for 90 days after entering the status 'SUBMITTED'.
4. The response will output the gzipped XML file that was originally uploaded as a partition.

3.4 Clarifying information on how to fill out key fields

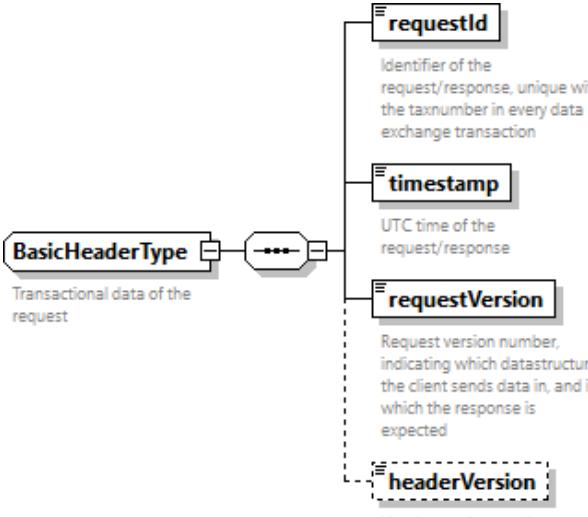
complexType BasicRequestType

<i>Diagram</i>	 <pre> sequenceDiagram participant BR as BasicRequestType participant H as header participant U as user BR->>H: BR-->>U: activate H activate U H->>U: deactivate H deactivate U </pre>
<i>Namespace</i>	http://schemas.nav.gov.hu/NTCA/1.0/common
<i>Tags</i>	<u>header</u> <u>user</u>
<i>It refers to</i>	complexType BasicEVatRequestType
<i>Content of</i>	Basic request data

element BasicRequestType/header

<i>Diagram</i>	 <pre> sequenceDiagram participant H as header participant BH as BasicHeaderType H->>BH: activate BH activate requestId activate timestamp activate requestVersion activate headerVersion deactivate requestVersion deactivate headerVersion deactivate timestamp deactivate requestId deactivate BH </pre>
<i>Namespace</i>	http://schemas.nav.gov.hu/NTCA/1.0/common
<i>Type</i>	BasicHeaderType
<i>Features</i>	content complex
<i>Tags</i>	<u>requestId</u> <u>timestamp</u> <u>requestVersion</u> <u>headerVersion</u>
<i>Content of</i>	Transactional data of the request

complexType BasicHeaderType

Diagram	
	 <pre> classDiagram class BasicHeaderType { requestId timestamp requestVersion headerVersion } BasicHeaderType < -- BasicRequestType BasicHeaderType < -- BasicResponseType </pre> <p>The diagram illustrates the structure of the BasicHeaderType. It consists of four attributes:</p> <ul style="list-style-type: none"> requestId: Identifier of the request/response, unique with the taxnumber in every data exchange transaction. timestamp: UTC time of the request/response. requestVersion: Request version number, indicating which datastructure the client sends data in, and in which the response is expected. headerVersion: Header version number (optional). <p>Transactional data of the request is indicated below the BasicHeaderType class.</p>
Namespace	http://schemas.nav.gov.hu/NTCA/1.0/common
Tags	requestId timestamp requestVersion headerVersion
It refers to	element BasicRequestType/header BasicResponseType/header
Content of	Transactional data of the request

Definition and related requirements

- **requestId** is the identification code of the request. **requestId** can be of any value that matches the pattern and does not violate the uniqueness constraint. The **requestId** must be unique for each request, with regards to the specific taxpayer and within the **timestamp** tolerance. This uniqueness constraint applies to all schema-valid requests, i.e. all successful requests and any validation error returned other than **INVALID_REQUEST**. The **requestSignature** value includes the tag value.

The **timestamp** contains the client-side time that the request was submitted. The **requestSignature** value includes the tag value. The **timestamp** in the request must be received in UTC time and the proper format. For Hungarian time zones, this translates to:

- GMT + 1 hour for DT (winter time)
- GMT + 2 hours for DST (daylight saving time)

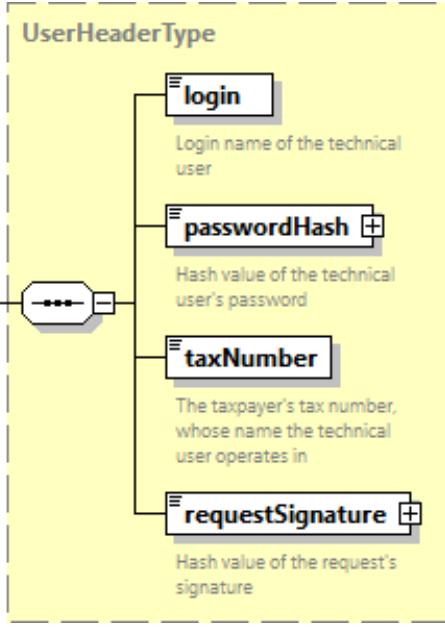
The allowed **timestamp** tolerance compared to the server time is ±1 day.

For more information on dates, please see the section “Converting local time to UTC”.

- **requestVersion** identifies the structure of the request. Subsequent interface modifications will refer back to this tag, meaning that **requestVersion** defines the structure of both the request and the response as well, as well as the related validations and verifications. Its value is to be filled in accordance with the value defined in the supported version. The value is checked through business validation, it has no XSD level enum value set. The only currently accepted value: 1.0
- **headerVersion** is an optional element of the request. It serves as the tag that subsequent structures and related verifications will refer back to, should request structures undergo any fundamental changes in the future. The value is checked through business validation, it has no XSD level enum value set. The only currently accepted value: 1.0



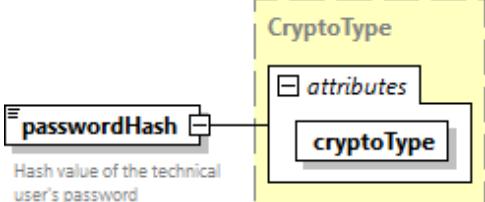
element **BasicRequestType/user**

<i>Diagram</i>	
<i>Namespace</i>	http://schemas.nav.gov.hu/NTCA/1.0/common
<i>Type</i>	<u>UserHeaderType</u>
<i>Features</i>	content complex
<i>Tags</i>	<u>login</u> <u>passwordHash</u> <u>taxNumber</u> <u>requestSignature</u>
<i>Content of</i>	Authentication data of the request

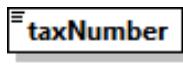
element **UserHeaderType/login**

<i>Diagram</i>													
<i>Namespace</i>	http://schemas.nav.gov.hu/NTCA/1.0/common												
<i>Type</i>	<u>LoginType</u>												
<i>Features</i>	content simple												
<i>Facets</i>	<table><thead><tr><th>Type</th><th>Value</th><th>Description</th></tr></thead><tbody><tr><td>minLength</td><td>6</td><td></td></tr><tr><td>maxLength</td><td>15</td><td></td></tr><tr><td>pattern</td><td>[a-zA-Z0-9]{6,15}</td><td></td></tr></tbody></table>	Type	Value	Description	minLength	6		maxLength	15		pattern	[a-zA-Z0-9]{6,15}	
Type	Value	Description											
minLength	6												
maxLength	15												
pattern	[a-zA-Z0-9]{6,15}												
<i>Content of</i>	Technical user's login name												

element **UserHeaderType/passwordHash**

<i>Diagram</i>	 <pre> classDiagram class passwordHash { <<Hash value of the technical user's password>> } class CryptoType { <<attributes>> cryptoType } passwordHash "1" -- "1" CryptoType : <<attributes>> </pre>								
<i>Namespace</i>	http://schemas.nav.gov.hu/NTCA/1.0/common								
<i>Type</i>	<u>CryptoType</u>								
<i>Features</i>	<i>content</i> complex								
<i>Facets</i>									
<i>Attributes</i>	<table> <thead> <tr> <th>Name</th> <th>Type</th> <th>Default value</th> <th>Annotations</th> </tr> </thead> <tbody> <tr> <td><u>cryptoType</u></td> <td><u>SimpleText</u></td> <td>50</td> <td><u>NotBlankType</u></td> </tr> </tbody> </table>	Name	Type	Default value	Annotations	<u>cryptoType</u>	<u>SimpleText</u>	50	<u>NotBlankType</u>
Name	Type	Default value	Annotations						
<u>cryptoType</u>	<u>SimpleText</u>	50	<u>NotBlankType</u>						
<i>Content of</i>	Technical user's password hash value								

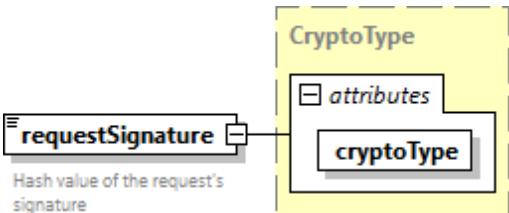
element **UserHeaderType/taxNumber**

<i>Diagram</i>	 <pre> classDiagram class taxNumber { <<The taxpayer's tax number, whose name the technical user operates in>> } </pre>															
<i>Namespace</i>	http://schemas.nav.gov.hu/NTCA/1.0/common															
<i>Type</i>	<u>TaxpayerIdType</u>															
<i>Features</i>	<i>content</i> simple															
<i>Facets</i>	<table> <thead> <tr> <th>Type</th> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>length</td> <td>8</td> <td></td> </tr> <tr> <td>minLength</td> <td>1</td> <td></td> </tr> <tr> <td>maxLength</td> <td>8</td> <td></td> </tr> <tr> <td>pattern</td> <td>[0-9]{8}</td> <td></td> </tr> </tbody> </table>	Type	Value	Description	length	8		minLength	1		maxLength	8		pattern	[0-9]{8}	
Type	Value	Description														
length	8															
minLength	1															
maxLength	8															
pattern	[0-9]{8}															
<i>Content of</i>	The tax number in the system belonging to the taxpayer on behalf of whom the technical user is acting															

element UserHeaderType/predecessorTaxnumber

<i>Diagram</i>	 <p>A jogelőd adószáma, aki nevében a technikai felhasználó tevékenykedik</p>																	
<i>Namespace</i>	http://schemas.nav.gov.hu/NTCA/1.0/common																	
<i>Type</i>	<u>TaxpayerIdType</u>																	
<i>Characteristics</i>	<p><i>minOcc</i> 0 <i>maxOcc</i> 1</p> <p><i>content</i> simple</p>																	
<i>Facets</i>	<table> <thead> <tr> <th><i>Type</i></th> <th><i>Value</i></th> <th><i>Description</i></th> </tr> </thead> <tbody> <tr> <td>length</td> <td>8</td> <td></td> </tr> <tr> <td>minLength</td> <td>1</td> <td></td> </tr> <tr> <td>maxLength</td> <td>8</td> <td></td> </tr> <tr> <td>pattern</td> <td>[0-9]{8}</td> <td></td> </tr> </tbody> </table>			<i>Type</i>	<i>Value</i>	<i>Description</i>	length	8		minLength	1		maxLength	8		pattern	[0-9]{8}	
<i>Type</i>	<i>Value</i>	<i>Description</i>																
length	8																	
minLength	1																	
maxLength	8																	
pattern	[0-9]{8}																	
<i>Contents</i>	Tax identification number of the predecessor entity represented by the technical user																	

element UserHeaderType/requestSignature

<i>Diagram</i>	 <p>Hash value of the request's signature</p>														
<i>Namespace</i>	http://schemas.nav.gov.hu/NTCA/1.0/common														
<i>Type</i>	<u>CryptoType</u>														
<i>Features</i>	<i>content</i> complex														
<i>Facets</i>	<table> <thead> <tr> <th><i>Type</i></th> <th><i>Value</i></th> <th><i>Description</i></th> </tr> </thead> <tbody> <tr> <td>minLength</td> <td>1</td> <td></td> </tr> <tr> <td>maxLength</td> <td>512</td> <td></td> </tr> <tr> <td>pattern</td> <td>.*[^\s].*</td> <td></td> </tr> </tbody> </table>			<i>Type</i>	<i>Value</i>	<i>Description</i>	minLength	1		maxLength	512		pattern	.*[^\s].*	
<i>Type</i>	<i>Value</i>	<i>Description</i>													
minLength	1														
maxLength	512														
pattern	.*[^\s].*														
<i>Attributes</i>	<table> <thead> <tr> <th><i>Name</i></th> <th><i>Type</i></th> <th><i>Default value</i></th> <th><i>Fixed Annotiations</i></th> </tr> </thead> <tbody> <tr> <td><u>cryptoType</u></td> <td><u>SimpleText50NotBlankType</u></td> <td></td> <td></td> </tr> </tbody> </table>			<i>Name</i>	<i>Type</i>	<i>Default value</i>	<i>Fixed Annotiations</i>	<u>cryptoType</u>	<u>SimpleText50NotBlankType</u>						
<i>Name</i>	<i>Type</i>	<i>Default value</i>	<i>Fixed Annotiations</i>												
<u>cryptoType</u>	<u>SimpleText50NotBlankType</u>														
<i>Content of</i>	Request signature hash value														



complexType **BasicResponseType**

<i>Diagram</i>	
<i>Namespace</i>	http://schemas.nav.gov.hu/NTCA/1.0/common
<i>Tags</i>	<u>header</u> <u>result</u>
<i>It refers to</i> <i>Type</i>	GeneralErrorHeaderResponseType ManageAttachmentUploadResponseType ManageDeclarationFinalizeResponseType ManageDeclarationPartitionResponseType ManageDeclarationSubmissionResponseType ManageDeclarationUploadResponseType PurgeAttachmentResponseType QueryAttachmentListResponseType QueryCustomsDeclarationDigestResponseType QueryCustomsDeclarationTaxCodeResponseType QueryDeclarationDataResponseType QueryDeclarationListResponseType QueryDeclarationProcessingStatusResponseType QueryDocumentListResponseType QueryDocumentListResultResponseType QueryInvoiceTaxCodeResponseType QueryTaxCodeCatalogResponseType QueryVatDeclarationDataResponseType
<i>Content of</i>	Basic response data

Definition and related requirements

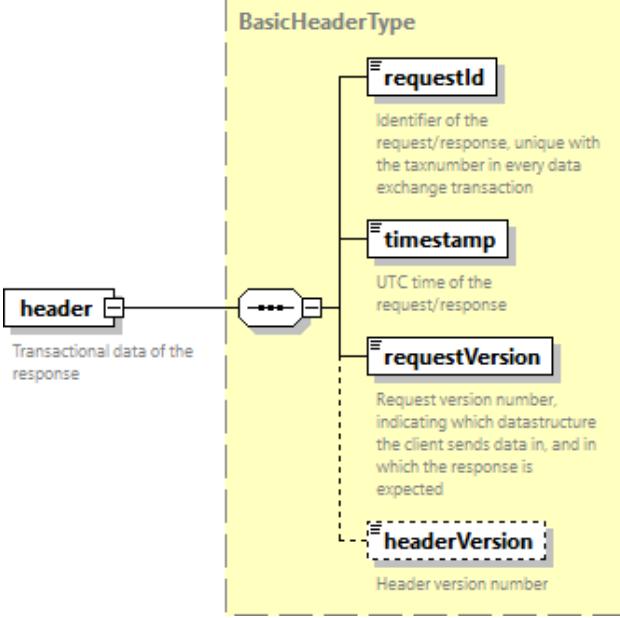
BasicResponseType is a mandatory part of every response element. The complex type and the header and result nodes defined within it are taken from common.xsd. Within the type, the header contains the transaction data for the response, and the result contains the result of processing.

In addition, the services return the Software node containing the invoicing program that performed the operation.

- funcCode is the status given by the server for executing the operation contained by the request. Its interpretation may vary depending on the business operation in question, but it should always be interpreted together with the full response.
- errorCode is returned when the value of the funcCode is ERROR. It contains the unique error code; this element can be used on the client side to map the error message. For more information on the errorCode value set, please consult the error code chart in the section “**ERROR MANAGEMENT**”.
- message is an optional text-based message attached to the funcCode or errorCode. It is a human-readable message, to aid comprehension.
- NAV will use the notification node to convey informational messages via API calls in the future, using a key-value structure.

In the responses, processing results are implemented by the *BasicResultType* element.

element BasicResponseType/header

<i>Diagram</i>	 <pre> classDiagram class BasicHeaderType { requestId : string timestamp : string requestVersion : string headerVersion : string } class header { <<Transactional data of the response>> } header --> BasicHeaderType </pre> <p>The diagram illustrates the structure of the BasicHeaderType. It is a class with four attributes: requestId, timestamp, requestVersion, and headerVersion. The requestId is described as an identifier of the request/response, unique with the taxnumber in every data exchange transaction. The timestamp is the UTC time of the request/response. The requestVersion is the request version number, indicating which datastructure the client sends data in, and in which the response is expected. The headerVersion is the header version number. A relationship exists between the header element (labeled "Transactional data of the response") and the BasicHeaderType class, indicated by a line connecting them.</p>
<i>Namespace</i>	http://schemas.nav.gov.hu/NTCA/1.0/common
<i>Type</i>	BasicHeaderType
<i>Features</i>	<i>content</i> complex
<i>Tags</i>	<u>requestId</u> <u>timestamp</u> <u>requestVersion</u> <u>headerVersion</u>
<i>Content of</i>	Transactional data of the response



element **BasicResponseType/result**

<i>Diagram</i>	<pre>graph LR; result --- <--> BasicResultType; subgraph BasicResultType [BasicResultType]; funcCode[funcCode
Processing result]; errorCode[errorCode
Processing error code]; message[message
Processing message]; notifications[notifications
Miscellaneous notifications]; end; result -.-> BasicResultType;</pre>
<i>Namespace</i>	http://schemas.nav.gov.hu/NTCA/1.0/common
<i>Type</i>	BasicResultType
<i>Features</i>	content complex
<i>Tags</i>	funcCode errorCode message notifications
<i>Content of</i>	Basic response result data

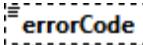
complexType **BasicResultType**

<i>Diagram</i>	<pre>graph LR; subgraph BasicResultType [BasicResultType]; funcCode[funcCode
Processing result]; errorCode[errorCode
Processing error code]; message[message
Processing message]; notifications[notifications
Miscellaneous notifications]; end; BasicResultType --- <--> boundary(()); boundary --- funcCode;</pre>
<i>Namespace</i>	http://schemas.nav.gov.hu/NTCA/1.0/common
<i>Tags</i>	funcCode errorCode message notifications
<i>It refers to</i>	element GeneralExceptionResponse BasicResponseType/result
<i>Content of</i>	Basic response result data

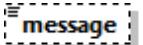
element **BasicResultType/funcCode**

<i>Diagram</i>	 funcCode Processing result
<i>Namespace</i>	http://schemas.nav.gov.hu/NTCA/1.0/common
<i>Type</i>	<u>FunctionCodeType</u>
<i>Features</i>	content simple
<i>Facets</i>	
<i>Content of</i>	Processing result

element **BasicResultType/errorCode**

<i>Diagram</i>	 errorCode Processing error code
<i>Namespace</i>	http://schemas.nav.gov.hu/NTCA/1.0/common
<i>Type</i>	<u>SimpleText50NotBlankType</u>
<i>Features</i>	
<i>Facets</i>	
<i>Content of</i>	Processing error code

element **BasicResultType/message**

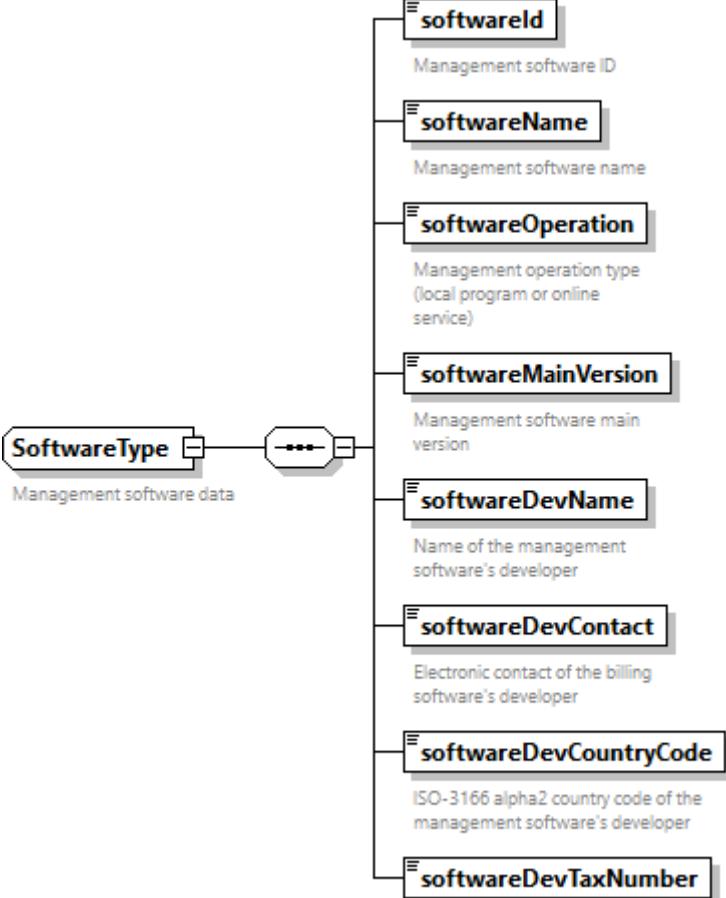
<i>Diagram</i>	 message Processing message
<i>Namespace</i>	http://schemas.nav.gov.hu/NTCA/1.0/common
<i>Type</i>	<u>SimpleText1024NotBlankType</u>

<i>Features</i>	
<i>Facets</i>	-
<i>Content of</i>	Processing message

element **BasicResultType/notifications**

<i>Diagram</i>	<pre> classDiagram class NotificationsType { notification [1..infinity] } notifications <--> notification </pre>
<i>Namespace</i>	http://schemas.nav.gov.hu/NTCA/1.0/common
<i>Type</i>	NotificationsType
<i>Features</i>	<p><i>minOcc</i> 0</p> <p><i>maxOcc</i> 1</p> <p><i>content</i> complex</p>
<i>Tags</i>	notification
<i>Content of</i>	Other notifications

complexType SoftwareType

Diagram  <pre> classDiagram class SoftwareType { *--> "7" "7" : softwareId "7" : softwareName "7" : softwareOperation "7" : softwareMainVersion "7" : softwareDevName "7" : softwareDevContact "7" : softwareDevCountryCode "7" : softwareDevTaxNumber } </pre>	softwareId Management software ID softwareName Management software name softwareOperation Management operation type (local program or online service) softwareMainVersion Management software main version softwareDevName Name of the management software's developer softwareDevContact Electronic contact of the billing software's developer softwareDevCountryCode ISO-3166 alpha2 country code of the management software's developer softwareDevTaxNumber Tax number of the management software's developer
<i>Namespace</i>	http://schemas.nav.gov.hu/EAR/1.0/api
<i>Tags</i>	softwareId softwareName softwareOperation softwareMainVersion softwareDevName softwareDevContact softwareDevCountryCode softwareDevTaxNumber
<i>It refers to</i>	<i>element</i> BasicEVatRequestType/software GeneralErrorResponseType/software
<i>Content of</i>	Management program data

Definition and related requirements

The types contains information regarding the software performing data reporting.

softwareId is an 18-length string used to identify the invoicing program used.

Recommendation for generating softwareId: The first two characters of the ID should be the country code of the software developer, as per ISO 3166 alpha-2. Additional characters of the identifier should include the tax registration number of the developer company, with the appropriate number of digits (in some countries, the length of the tax registration number may differ significantly from the 8 digits used in Hungary).

Additional characters of the identifier shall be generated by the developer in such a way as to ensure the identifier's uniqueness. The developer may decide to create separate identifiers for different versions of a particular software



product, or for instances running at different customers. The same instance of the same software version must report the same softwareId.

If multiple software development companies are involved in the development of the management program (or its individual modules), the identifier should include the data of one of them, based on mutual agreement.

The softwareId and the email address of the developer company (softwareDevContact) are used to identify potential operational problems, and to notify the developer.

The accurate filling of softwareDevTaxNumber is recommended, especially for Hungarian developers. Namely, NAV places any indications supporting data correctness in the electronic repository of taxpayers identifiable by tax number. For this reason, it is worth for the software developer to review this piece of data in the invoicing program so that in the future, the developer can also be contacted instead of the taxpayer with a specific error in a given program.

4 ERROR MANAGEMENT

The service operates on a common list of values and error codes from the value set enumerated on the server side. Unlike the result codes, error codes intentionally do not appear in the enumerations of the schema definition, so that their potential change or expansion shall not cause implementation dependency on the client side. The result codes can be returned in the funcCode tag of the BasicResultType node, whereas the error codes can be returned in the errorCode tag in the return message. The returned funcCode values should be interpreted in accordance with the requested business process.

4.1 General error codes

4.1.1 GeneralExceptionResponseType

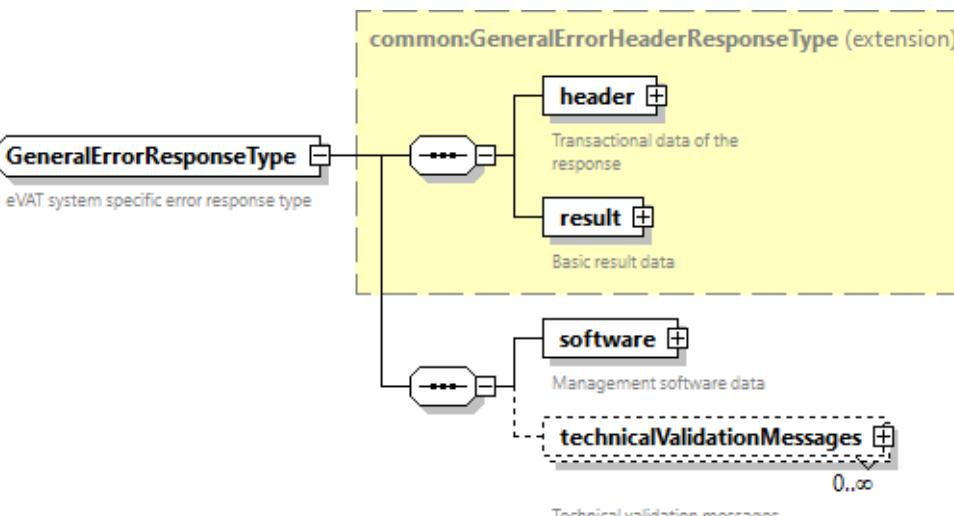
The response to technically unprocessable messages (non well-formed XML, incorrect namespace or incorrect context root) will contain a *GeneralExceptionResponseType* type error message for every operation of the service.

In this case, *notificationCode = SCHEMA_VIOLATION* will be set. This type extends *BasicResultType*, but contains no other element in addition to the ones contained in it.

The *notification* list type contains an itemised list of all schema violations related to the *earApi* schema definition if the request contained at least 1 tag that is not schema valid.

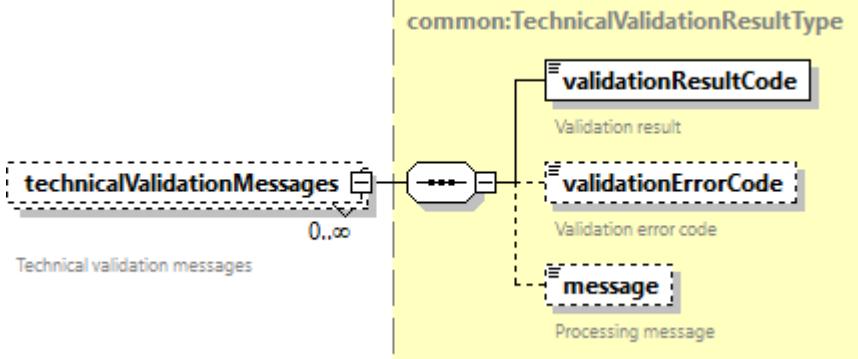
4.1.2 GeneralErrorResponseType

complexType GeneralErrorResponseType

<i>Diagram</i>	 <pre> classDiagram GeneralErrorResponseType < -- GeneralErrorResponseType GeneralErrorResponseType < -- common:GeneralErrorHeaderResponseType GeneralErrorResponseType --> header :+ GeneralErrorResponseType --> result :+ GeneralErrorResponseType --> software :+ GeneralErrorResponseType --> technicalValidationMessages :+ 0..∞ </pre>
<i>NameSpace</i>	http://schemas.nav.gov.hu/EAR/1.0/api
<i>Type</i>	<u>GeneralErrorHeaderResponseType</u> extension
<i>Features</i>	<i>base</i> common:GeneralErrorHeaderResponseType
<i>Tags</i>	<u>header</u> <u>result</u> <u>software</u> <u>technicalValidationMessages</u>
<i>It refers to</i>	element <u>GeneralErrorResponse</u>

Content of Type of error response specific to the eVAT system

element **GeneralErrorResponseType/technicalValidationMessages**

<i>Diagram</i>	 <pre> classDiagram common::TechnicalValidationResultType { validationResultCode validationErrorCode message } technicalValidationMessages < -- common::TechnicalValidationResultType technicalValidationMessages *--> common::TechnicalValidationResultType : 0..∞ </pre>
<i>Namespace</i>	http://schemas.nav.gov.hu/EAR/1.0/api
<i>Type</i>	<u>TechnicalValidationResultType</u>
<i>Features</i>	<p><i>minOcc</i> 0</p> <p><i>maxOcc</i> unbounded</p> <p><i>content</i> complex</p>
<i>Tags</i>	<u>validationResultCode</u> <u>validationErrorCode</u> <u>message</u>
<i>Content of</i>	Technical validation messages

Definition and related requirements

The general error type message for each operation of the service is implemented by *GeneralErrorResponseType*.

This type extends *GeneralErrorHeaderResponseType*, meaning that in addition to the elements it contains, it will also incorporate a technical validation list type.

The technical error codes detailed in the next section are all returned in the *GeneralExceptionResponse* or the *GeneralErrorResponse* response element. The response elements detailed in the section “Error management” are generated only if the synchronous processing has been validated on both the technical and business logic ends. Thus, any error tag received in the HTTP response body will always indicate some kind of error.

- 1) If a technicalValidationMessages tag is generated, the value of the validationResultCode element can only be ERROR (CRITICAL is a value reserved in this type for potential future validations).
- 2) The validationErrorCode tag contains the error type code.
- 3) For schema violations, the message tag contains the name, value and expected value of the incorrect tag that failed technical validation; in other cases, it contains a text-based error message for the validationErrorCode tag

4.2 Technical error codes

The errorCode value set of the synchronous calls is listed in the following table.

Technical and authentication errors

#	http response	Response body	funcCode	errorCode	requestVersion
1	HTTP 404 NOT_FOUND	-	-	-	1.0
2	HTTP 405 METHOD_NOT_ALLOWED	GeneralExceptionResponse XML tag	ERROR	NOT_ALLOWED_EXCEPTION	1.0
3	HTTP 400 BAD_REQUEST	GeneralExceptionResponse XML tag	ERROR	INVALID_REQUEST	1.0
4	HTTP 400 BAD_REQUEST	GeneralExceptionResponse XML tag	ERROR	INVALID_REQUEST	1.0
5	HTTP 401 UNAUTHORIZED	GeneralErrorResponse XML tag	ERROR	INVALID_SECURITY_USER	1.0
6	HTTP 500 INTERNAL_SERVER_ERROR	GeneralErrorResponse XML tag	ERROR	NOT_REGISTERED_CUSTOMER	1.0
7	HTTP 500 INTERNAL_SERVER_ERROR	GeneralErrorResponse XML tag	ERROR	INVALID_CUSTOMER	1.0
8	HTTP 500 INTERNAL_SERVER_ERROR	GeneralErrorResponse XML tag	ERROR	INVALID_USER_RELATION	1.0
9	HTTP 403 FORBIDDEN	GeneralErrorResponse XML tag	ERROR	FORBIDDEN	1.0
10	HTTP 400 BAD_REQUEST	GeneralErrorResponse XML tag	ERROR	REQUEST_ID_NOT_UNIQUE	1.0
11	HTTP 400 BAD_REQUEST	GeneralErrorResponse XML tag	ERROR	INVALID_REQUEST_SIGNATURE	1.0



12	HTTP SERVICE_UNAVAILABLE	503	GeneralErrorResponse XML tag	ERROR	SERVICE_UNAVAILABLE	1.0
13	HTTP 400 BAD_REQUEST		GeneralErrorResponse XML tag	ERROR	INVALID_TIMESTAMP	1.0
14	HTTP 400 BAD_REQUEST		GeneralErrorResponse XML tag	ERROR	INVALID_PASSWORD_HASH_CRYPTO	1.0
15	HTTP 400 BAD_REQUEST		GeneralErrorResponse XML tag	ERROR	INVALID_REQUEST_SIGNATURE_HASH_CRYPTO	1.0
16	HTTP 400 BAD_REQUEST		GeneralErrorResponse XML tag	ERROR	INVALID_REQUEST_VERSION	1.0
17	HTTP 400 BAD_REQUEST		GeneralErrorResponse XML tag	ERROR	INVALID_HEADER_VERSION	1.0
18	HTTP UNSUPPORTED_MEDIA_TYPE	415	GeneralExceptionResponse XML tag	ERROR	INVALID_REQUEST	1.0
19	HTTP 416 NOT_ACCEPTABLE		GeneralExceptionResponse XML tag	ERROR	INVALID_REQUEST	1.0
20*	HTTP 400 BAD_REQUEST		GeneralErrorResponse XML tag	ERROR	REQUEST_VERSION_NOT_ALLOWED	2.0
21*	HTTP TOO_MANY_REQUESTS	429	GeneralErrorResponse XML tag	ERROR	TOO_MANY_REQUESTS	1.0
22	HTTP 400 BAD_REQUEST		GeneralErrorResponse	ERROR	PARTITIONCOUNT_LIMIT_EXCEEDED	1.0
23	HTTP 400 BAD_REQUEST		GeneralErrorResponse	ERROR	INVALID_DECLARATION_UPLOAD_ID	1.0
24	HTTP 400 BAD_REQUEST		GeneralErrorResponse	ERROR	PARTITION_ALREADY_EXISTS	1.0
25	HTTP 400 BAD_REQUEST		GeneralErrorResponse	ERROR	PARTITION_SIZE_LIMIT_EXCEEDED	1.0
26	HTTP 400 BAD_REQUEST		GeneralErrorResponse	ERROR	FILE_COUNT_LIMIT_EXCEEDED	1.0
27	HTTP 400 BAD_REQUEST		GeneralErrorResponse	ERROR	FILE_SIZE_LIMIT_EXCEEDED	1.0
28	HTTP 400 BAD_REQUEST		GeneralErrorResponse	ERROR	DISK_QUOTA_EXCEEDED	1.0



29	HTTP 400 BAD_REQUEST	GeneralErrorResponse	ERROR	FILE_TYPE_MISMATCH	1.0
30	HTTP 400 BAD_REQUEST	GeneralErrorResponse	ERROR	UNSUPPORTED_HASH_TYPE	1.0
31	HTTP 400 BAD_REQUEST	GeneralErrorResponse	ERROR	INVALID_FILE_HASH	1.0
32	HTTP 400 BAD_REQUEST	GeneralErrorResponse	ERROR	DECLARATION_PARTITION_NOT_FOUND	1.0
33	HTTP 400 BAD_REQUEST	GeneralErrorResponse	ERROR	PRELIMINARY_CONFIRMATION_NOT_FALSE	1.0
34	HTTP 400 BAD_REQUEST	GeneralErrorResponse	ERROR	INVALID_DECLARATIONPROCESSINGID	1.0
35	HTTP 400 BAD_REQUEST	GeneralErrorResponse	ERROR	INVALID_STATUS	1.0
36	HTTP 400 BAD_REQUEST	GeneralErrorResponse	ERROR	SUBMISSION_PERIOD_CLOSED	1.0
37	HTTP 400 BAD_REQUEST	GeneralErrorResponse	ERROR	INVALID_CLAIMCHECKID	1.0
38	HTTP 400 BAD_REQUEST	GeneralErrorResponse	ERROR	PROTECTED_CLAIMCHECKID	1.0
39	HTTP 500 INTERNAL_SERVER_ERROR	GeneralErrorResponse	ERROR	INCORRECT_XSD_VERSION	1.0
40	HTTP 500 INTERNAL_SERVER_ERROR	GeneralErrorResponse	ERROR	CONFLICTING_XSD_VALIDITY_VERSION	1.0
41	HTTP 500 INTERNAL_SERVER_ERROR	GeneralErrorResponse	ERROR	CONFLICTING_PERIOD_XSD_VERSION	1.0

*Error codes of validations to be introduced in the future.

Error, actions

#	errorCode	Cause of error	Action
1	-	incorrect service endpoint in request	For more information on addressing endpoints in various environments, see the section “Environment Accessibility”; check the URL.
2	NOT_ALLOWED_EXCEPTION	incorrect http method in request	The URL is correct, but the http method is not POST. All operations of the interface must be sent with POST method.



3	INVALID_REQUEST	non well-formed in XML in request body	A syntactically incorrect XML message cannot be considered XML or processed as such, as per the XML standard, and must be corrected.
4	INVALID_REQUEST	not schema-valid XML in request body	The elements of the entered XML listed in the response violate the restrictions of the earApi.xsd, and must be corrected. (The error code may also show up if a request is not sent to the endpoint of the appropriate operation)
5	INVALID_SECURITY_USER	incorrect login + passwordHash in the request	This error message may be returned in various cases. Possible causes: no user under the entered login name, incorrect password, the login + passwordHash pair is semantically correct, but the password hash was not calculated correctly on the client side. The correctness of data and hashes should be checked; if necessary, the taxpayer using the technical user should be contacted.
6	NOT_REGISTERED_CUSTOMER	the taxpayer specified in the request is not found	No taxpayer using the system with the tax number included in the user tag can be found.
7	INVALID_CUSTOMER	incorrect taxNumber in the request	The tax number in the user tag does not exist, or its status does not allow for performing actions. The correctness of data should be checked; if necessary, the concerned taxpayer should be contacted.
8	INVALID_USER_RELATION	no connection between the entities in the request	There is no technical user with the entered login name and tax number, or the user status does not allow for performing that action any longer. The correctness of data should be checked; if necessary, the concerned taxpayer should be contacted.
9	FORBIDDEN	the technical user in the request does not have the rights to query the endpoint service	The primary users of the taxpayer allocate the rights to technical users. The concerned taxpayer should be contacted, if necessary.
10	REQUEST_ID_NOT_UNIQUE	the requestId in the request is not unique	The requestId for the tax number in the request has already been used. A new ID meeting the uniqueness requirement should be provided.
11	INVALID_REQUEST_SIGNATURE	incorrect requestSignature in the request	The requestSignature calculation on the server side does not correspond with the value calculated on the client side. For more information about the calculation, see the section “Calculating requestSignature”.
12	SERVICE_UNAVAILABLE	maintenance in progress	The requested operation is temporarily suspended due to maintenance. Furthermore, the error can be returned even if the Gateway or the underlying specialized system is undergoing maintenance. Please monitor the information on the interface, and repeat your request at a later time.



13	INVALID_TIMESTAMP	the timestamp indicated in the query is more than 1 day old	The error message can be returned in the operation containing all the authentication if the value of the timestamp indicated in the header node is outside the ± 1 day interval of the server time.
14	INVALID_PASSWORD_HASH_CRYPTO_TYPE	incorrect hashing algorithm for the technical user's password	This error message can be returned if the value of user/passwordHash/cryptoType is not SHA-512.
15	INVALID_REQUEST_SIGNATURE_HASH_CRYPTO	incorrect hashing algorithm for the request signature	This error message can be returned if the value of user/requestSignature/cryptoType is not SHA3-512.
16	INVALID_REQUEST_VERSION	request version is invalid	This error message can be returned if the value of header/requestVersion is not 1.0
17	INVALID_HEADER_VERSION	header version is invalid	This error message can be returned if the value of /header/headerVersion is specified and not equal to 1.0
18	INVALID_REQUEST	the request was not submitted with the correct Content-type header	The request was not submitted with the correct media type for the operation. The request media type and/or the Content-type header value should be corrected.
19	INVALID_REQUEST	the request was not submitted with the correct Accept header	The server cannot return a response matching the Accept header in the request. The value of the Accept header needs to be corrected.
20	REQUEST_VERSION_NOT_ALLOWED	the value of the requestVersion tag in the request is no longer allowed	The requestVersion value of the request is a version that is no longer supported. This may occur when a newer version of the interface must be used due to a change in rules, where the earlier version can no longer be used from a given time. Must be corrected!
21	TOO_MANY_REQUESTS	too many requests submitted within a short time	This error message can be returned if the taxpayer has submitted too many requests within a given time window. In these cases, requests will be rejected in order to preserve resources. After the time window has expired, the transfer of requests may be resumed.
22	PARTITIONCOUNT_LIMIT_EXCEEDED	Maximum partition count overrun!	The number of partitions of the declaration exceeds the allowed quantity. The number of partitions needs to be reduced.
23	INVALID_DECLARATION_UPLOAD_ID	The partition has exceeded the size limit!	The ID referenced is not valid. The upload ID has a validity period which may have expired. A new ID must be requested.



24	PARTITION_ALREADY_EXISTS	The partition already exists!	The partition you want to upload has already been uploaded. If you want to update the partition, you will need to request a new upload ID; it is not possible to replace an existing partition that has already been uploaded.
25	PARTITION_SIZE_LIMIT_EXCEEDED	The partition has exceeded the size limit!	The partition size needs to be reduced.
26	FILE_COUNT_LIMIT_EXCEEDED	The number of uploadable files exceeded!	The maximum number of files allowed to be stored in the repository has been exceeded. It is recommended that you delete files that are no longer needed from the repository.
27	FILE_SIZE_LIMIT_EXCEEDED	The file has exceeded the size limit!	The file is too large, it is recommended to reduce the file size.
28	DISK_QUOTA_EXCEEDED	Storage space quota overrun!	The available storage quota is full. It is recommended that you delete files that are no longer needed from the repository.
29	FILE_TYPE_MISMATCH	File type mismatch!	The specified file type and the metadata of the uploaded file do not match.
30	UNSUPPORTED_HASH_TYPE	Unsupported hashtype!	The specified hashtype is not supported. Currently, SHA3-512 should be used.
31	INVALID_FILE_HASH	File hash mismatch!	The hash of the uploaded file and the hash value submitted in the request do not match.
32	DECLARATION_PARTITION_NOT_FOUND	Partition not found: {list of missing partitions}	The upload cannot be finalised due to missing partitions. It is recommended to upload them and retry the finalize call.
33	PRELIMINARY_CONFIRMATION_NOT_FALSE	The value of prior approval is not false!	The field value must be set to false.
34	INVALID_DECLARATIONPROCESSINGID	Invalid processing ID!	The processing ID was entered incorrectly. Must be corrected!
35	INVALID_STATUS	The status of the analytics does not match the status required for submission!	<p>The status of the declaration is not “FINISHED”. This may have several reasons:</p> <ul style="list-style-type: none">• processing is still in progress and is in <i>PROCESSING</i> or <i>BEVFELD_CHECK</i> status• the declaration is in <i>ABORTED</i> status and cannot be submitted, in which case the corrected declaration must be uploaded again
36	INVALID_CLAIMCHECKID	Invalid claimCheckId!	The attachment with the given identifier is not found. The request must be corrected!



37	PROTECTED_CLAIMCHECKID	The uploaded file cannot be deleted!	The attachment is in a non-deletable state, due to the processing of the declaration. Once the declaration has been processed, it becomes deletable again.		
----	------------------------	--------------------------------------	--	--	--

Errors in processing

#	HTTP response	Response body	funcCode	errorCode	request Version
1	HTTP 500 INTERNAL_SERVER_ERR OR	GeneralErrorResponse	ERROR	OPERATION_FAILED	1.0

Error, actions

#	errorCode	Cause of error	Action
1	OPERATION_FAILED	unexpected processing error	<p>Server-side general error code. This error can occur only for synchronous requests; try repeating the operation after a short wait. If the operation is repeatedly unsuccessful in a live environment, contact the NAV Helpdesk. However, check the website first to make sure that there is no system outage or transient malfunction causing the error. Please note that there is no guaranteed availability in the user test environment; therefore, any errors occurring in the test system should not be reported.</p> <p>(The error code may also show up if a request is not sent to the endpoint of the appropriate operation)</p>

4.3 Validation error codes

4.3.1 Technical validation and blocking validation error codes

If the submitted analytics is not schema-valid or contains other errors, this fact will be returned in this branch. These errors and their suggested solutions are listed in the section “ERROR type errors” of Annex I. With an error of this type, the declaration cannot be submitted, and enters the status ABORTED.

4.3.2 Warnings

Warnings do not prevent the submission of the declaration, however, they suggest that the content of the declaration should be reviewed and corrected, if necessary. The warnings are typed, the concerned tags for which warnings have been issued are indicated in the message as many times as the data report violates the restriction defined in the validation.

The processing system generates and includes the warnings in the response messages to inform the taxpayer. The purpose of the notification regarding errors detected in the analytics which do not render the submission of the declaration impossible is to help identify and correct software, data input errors, and possible faulty practices on the part of the taxpayer. Warnings are only issued if the problem exists beyond any doubt – if there is any level of doubt, no warning is generated.

The fact that no alert is generated in a piece of analytics when it is received cannot be construed as legal or technical proof that the data provided is correct. An automated test will not necessarily detect a data error in the data report, particularly with respect to discrepancies between the original document and the content of the data report.

The list of data and contexts checked as part of processing the analytics remains open, and may be continuously extended based on the experiences with the system.

There is no mandated procedure by law, therefore no general policy exists about what exact actions should be taken in response to warnings. The best practice is essentially determined by the nature of the situation warranting a warning – software errors, data input errors, faulty practices. As a wide variety of different situations can trigger warnings, it might be useful to make the warnings available to the users, the staff responsible for accounting, the technical support unit, the developer of the software – or even several of these, depending on the nature, business and staff size of the company.

The detailed description of warnings can be found in the section of Annex I containing the WARNING and INFO messages.

5 ENVIRONMENTAL AVAILABILITY

The eVAT M2M system services are available in the following environments and at the following addresses.

5.1 User test environment

eVAT interface: <https://api-test.eafa.nav.gov.hu>

URLs and resources:

- <https://api-test.eafa.nav.gov.hu/analyticsService/v1/manageDeclarationUpload>
- <https://api-test.eafa.nav.gov.hu/analyticsService/v1/manageDeclarationPartition>
- <https://api-test.eafa.nav.gov.hu/analyticsService/v1/manageDeclarationFinalize>
- <https://api-test.eafa.nav.gov.hu/analyticsService/v1/queryDeclarationProcessingStatus>
- <https://api-test.eafa.nav.gov.hu/analyticsService/v1/manageDeclarationSubmission>
- <https://api-test.eafa.nav.gov.hu/analyticsService/v1/queryCustomsDeclarationDigest>
- <https://api-test.eafa.nav.gov.hu/analyticsService/v1/queryInvoiceTaxCode>
- <https://api-test.eafa.nav.gov.hu/analyticsService/v1/queryCustomsDeclarationTaxCode>
- <https://api-test.eafa.nav.gov.hu/analyticsService/v1/queryDocumentList>
- <https://api-test.eafa.nav.gov.hu/analyticsService/v1/queryDocumentListResult>
- <https://api-test.eafa.nav.gov.hu/analyticsService/v1/queryDeclarationList>
- <https://api-test.eafa.nav.gov.hu/analyticsService/v1/queryDeclarationData>
- <https://api-test.eafa.nav.gov.hu/analyticsService/v1/queryTaxCodeCatalog>
- <https://api-test.eafa.nav.gov.hu/analyticsService/v1/queryVatDeclarationData>
- <https://api-test.eafa.nav.gov.hu/analyticsService/v1/manageAttachmentUpload>
- <https://api-test.eafa.nav.gov.hu/analyticsService/v1/queryAttachmentList>
- <https://api-test.eafa.nav.gov.hu/analyticsService/v1/purgeAttachment>

Swagger (openapi) access: <https://api-test.eafa.nav.gov.hu/v1/xmlapi/public/openapi/>

5.2 Live environment

eVAT interface: <https://api.eafa.nav.gov.hu>

URLs and resources:

- <https://api.eafa.nav.gov.hu/analyticsService/v1/manageDeclarationUpload>
- <https://api.eafa.nav.gov.hu/analyticsService/v1/manageDeclarationPartition>
- <https://api.eafa.nav.gov.hu/analyticsService/v1/manageDeclarationFinalize>
- <https://api.eafa.nav.gov.hu/analyticsService/v1/queryDeclarationProcessingStatus>
- <https://api.eafa.nav.gov.hu/analyticsService/v1/manageDeclarationSubmission>
- <https://api.eafa.nav.gov.hu/analyticsService/v1/queryCustomsDeclarationDigest>
- <https://api.eafa.nav.gov.hu/analyticsService/v1/queryInvoiceTaxCode>
- <https://api.eafa.nav.gov.hu/analyticsService/v1/queryCustomsDeclarationTaxCode>
- <https://api.eafa.nav.gov.hu/analyticsService/v1/queryDocumentList>
- <https://api.eafa.nav.gov.hu/analyticsService/v1/queryDocumentListResult>
- <https://api.eafa.nav.gov.hu/analyticsService/v1/queryDeclarationList>
- <https://api.eafa.nav.gov.hu/analyticsService/v1/queryDeclarationData>

- <https://api.eafa.nav.gov.hu/analyticsService/v1/queryTaxCodeCatalog>
- <https://api.eafa.nav.gov.hu/analyticsService/v1/queryVatDeclarationData>
- <https://api.eafa.nav.gov.hu/analyticsService/v1/manageAttachmentUpload>
- <https://api.eafa.nav.gov.hu/analyticsService/v1/queryAttachmentList>
- <https://api.eafa.nav.gov.hu/analyticsService/v1/purgeAttachment>

Swagger (openapi) access: <https://api.eafa.nav.gov.hu/v1/xmlapi/public/openapi/>

6 HELPDESK AND TECHNICAL SUPPORT

This section provides information on troubleshooting and further support.

Tools assisting implementation checking

You can find more information about verifying certain codes and hashes as well as the general format of XML syntax in the following websites.

- Current UTC central time: <https://www.timeanddate.com/worldclock/timezone/utc>
- BASE64 online encode/decode: <https://www.base64decode.org/>
- Online CRC calculation: <https://www.functions-online.com/crc32.html> (online converters typically use hexadecimal values – these are acceptable, but in such cases the output must be converted to decimal value before use)
- SHA-512 online encode: <http://www.convertstring.com/Hash/SHA512>
- SHA3-512 online encode: <https://codebeautify.org/sha3-512-hash-generator>
- AES-128 ECB online decode: <https://8gwifi.org/CipherFunctions.jsp> (Select the AES ECB PKCS5PADDING option)
- Online validation tool for XML formatting and schema conformance: <https://www.xmlvalidation.com/>
- Regex validation: <https://regex101.com/>
- XML syntax information: https://www.w3schools.com/xml/xml_syntax.asp
- XML schema information: https://www.w3schools.com/xml/schema_intro.asp

6.1 Contact the Helpdesk

Two separate helpdesks are available to solve problems and answer questions regarding the eVAT system. Regarding questions and problems about the live system, please send a message via https://nav.gov.hu/ugyfeliranytu/keressen_minket/levelkuldes with the subject line ***"IT errors affecting the machine-to-machine interface of the eVAT system"*** ("eVAT, IT problems"). The form is also available in English.

Technical support for developers regarding exclusively the TEST system is available upon request sent to the e-mail address init.eafa_teszt_support@nav.gov.hu.

If you need technical support for using the interface, please include the content of the complete HTTP request (header and body) as well as the date of submission.

6.2 Availability on Github

<https://github.com/nav-gov-hu/eVAT>

6.2.1 Common repository

This repository was created for the version management of atomic types, business-catalog-type elements, and generic API communication types in a separate, common XSD (common.xsd). This allows these schema elements to be used by multiple NAV projects, thereby unifying API communication.

Repository available at: <https://github.com/nav-gov-hu/Common>

7 SYSTEM DIAGNOSTICS

8 ANNEXES

- I. Warning messages sent by the eVAT M2M system

I. WARNING MESSAGES SENT BY THE EVATM2M SYSTEM

ERROR type errors

ID	-
Error code	CONFLICTING_INDEX_METHOD
Error message	The ordering index in the submitted declaration is not associated with the correct declaration type.
Node	-
Condition	The amendment ordering for the declarations is not correct for the type of the submitted declaration and the flag for the correction requested by NAV.
Comment	-

ID	-
Error code	CONFLICTING_METHOD_SUCCESSION
Error message	The submitted declaration type is incorrect.
Node	-
Condition	After the submission of a basic, self-audit, corrigendum or correction-type declaration, the next declaration submitted was of an incorrect type.
Comment	-

ID	-
Error code	CONFLICTING_REPAIRMETHOD
Error message	The type of the corrective declaration does not match the type of the declaration to be corrected.
Node	-
Condition	The value of the flag for the correction requested by NAV indicates that the declaration was filed for corrective purposes, but the type of the corrective declaration does not match the type of the declaration to be corrected.
Comment	-

ID	-
Error code	MISSING_MODIFICATION_REFERENCE
Error message	For corrective declarations, it is mandatory to fill in the data in the amendment node.
Node	-
Condition	When correcting a previously submitted declaration, the amendment node and its data are missing.
Comment	-

ID	-
Error code	MISSING_DECLARATION
Error message	No previously submitted declaration was found with the given ID.
Node	-
Condition	The ID for the previously submitted declaration references by a self-audit, corrigendum or correction does not exist, or does not belong to the taxpayer submitting the declaration.
Comment	-

ID	-
Error code	INCORRECT_DECLARATION_STATUS
Error message	The status of the self-audited declaration is incorrect.
Node	-
Condition	The declaration affected by the amendment is in a status for which no self-audit, corrigendum or correction can be submitted.
Comment	-

ID	-
Error code	CONFLINCTING_DECLARATION_INDEX
Error message	The ordering of the amendments to the submitted declaration is incorrect.
Node	-
Condition	The ordering of the submitted declaration filed is not a monotonically increasing integer.
Comment	-

ID	-
Error code	CONFLICTING_PERIOD_START
Error message	The value of periodStart received in the ManageDeclarationUpload request does not match the value of vatDeclarationData/declarationInfo/declarationPeriodStart received in the XML.
Node	-
Condition	The start of the declaration period indicated when initiating the upload of the analytics does not correspond to the start of the declaration period indicated in the declaration file data.
Comment	-

ID	-
Error code	CONFLICTING_PERIOD_END
Error message	The value of periodEnd received in the ManageDeclarationUpload request does not match the value of vatDeclarationData/declarationInfo/declarationPeriodEnd received in the XML.
Node	-
Condition	The end the declaration period indicated when initiating the upload of the analytics does not correspond to the end of the declaration period indicated in the declaration file data.
Comment	-

ID	C0001
Error code	INVALID_XML_HASH
Error message	Invalid XML hash!
Node	-
Condition	The hash of the uploaded declaration does not match the hash previously provided.
Comment	-

ID	C0002
Error code	INVALID_XML_SHEMA
Error message	Invalid XML schema!
Node	-
Condition	The uploaded declaration is not schema valid according to earData.xsd.
Comment	-

ID	C0003
Error code	OPERATION_FAILED
Error message	Processing failed!
Node	-
Condition	An error occurred during processing.
Comment	-

ID	C0004
Error code	XML_FILESIZE_TOO_LARGE
Error message	The XML file size is too large!
Node	-
Condition	The decompressed size of the uploaded declaration exceeds 2GB.
Comment	-

ID	C0005
Error code	FILE_EXTRACTION_FAILED
Error message	File unpacking failed!
Node	-
Condition	The uploaded file is not properly compressed.
Comment	-

ID	G0002
Error code	INCORRECT_TAXNUMBER
Error message	Incorrect tax number
Node	declarationInfo/taxNumber
Condition	declarationInfo/taxNumber does not match the tax number of the declaration submitter
Comment	

ID	G0003
Error code	NOT_ALLOWED_DECLARATIONTYPE
Error message	Filling the declaration type is not allowed.
Node	declarationInfo/declarationType
Condition	declarationInfo/declarationType != NONE
Comment	

ID	G0004
Error code	NOT_ALLOWED_DECLARATIONKIND
Error message	Filling the declaration kind is not allowed.
Node	declarationInfo/declarationKind
Condition	declarationInfo/declarationKind != NONE
Comment	

ID	G0005
Error code	CONFLICTING_FREQUENCY_PERIOD
Error message	The declaration frequency provided does not match the declaration frequency in the master data!
Node	DeclarationStatements/ProcedureStatements
Condition	declarationInfo.declarationFrequency != declaration frequency from the taxpayer's master data AND thresholdExceededIndicator = false AND interimFrequencyChangeIndicator =false AND midYearCommunityTaxNumberIndicator = false AND outOfTurnDeclarationCode = null AND becomesLiveAgainIndicator = false
Comment	

ID	G0007
Error code	CONFLICTING_RETURNDECISION_RETURNSTATEMENTS
Error message	As per the transfer decision, you do not request a transfer, but the transfer statement has been completed nonetheless.
Node	declarationStatements/returnDecision
Condition	declarationStatements/returnDecision == "NO_RETURN" AND declarationStatements/returnStatements node exists
Comment	

ID	G0008
Error code	CONFLICTING_RETURNDECISION_RETURNSTATEMENTS
Error message	Despite the transfer decision, there is no transfer statement.
Node	declarationStatements/returnDecision
Condition	declarationStatements/returnDecision != "NO_RETURN" AND declarationStatements/returnStatements node does not exist.
Comment	

ID	G0009
Error code	CONFLICTING_RETURNSTATEMENTS_SUMDEDUCTIBLETAX
Error message	Missing return statement. Based on the declaration frequency and the amount of tax refundable, the return statement is missing.
Node	declarationStatements/returnStatements
Condition	declarationStatements/returnStatements node does not exist AND declarationSummary/sumDeductibleTax > 1000 AND declarationInfo/declarationFrequency == "MONTHLY" OR declarationSummary/sumDeductibleTax > 1000 AND declarationInfo/declarationFrequency == "MONTHLY" OR declarationSummary/sumDeductibleTax > 250 AND declarationInfo/declarationFrequency == "QUARTERLY" OR declarationSummary/sumDeductibleTax > 50 AND declarationInfo/declarationFrequency == "ANNUAL". Values are in thousands of forints.
Comment	

ID	G0010
Error code	INCORRECT_SUMPAYABLETAX
Error message	The amount of tax payable cannot be a negative number.
Node	declarationSummary/sumPayableTax
Condition	declarationSummary/sumPayableTax < 0.
Comment	

ID	G0011
Error code	INCORRECT_SUMDEDUCTIBLETAX
Error message	The amount of tax refundable cannot be a negative number.
Node	declarationSummary/sumDeductibleTax
Condition	declarationSummary/sumDeductibleTax < 0.
Comment	

ID	G0012
Error code	INCORRECT_SUMRESIDUALTAX
Error message	The amount of decreasing item available for setoff from the previous period cannot be negative.
Node	declarationSummary/sumResidualTax
Condition	declarationSummary/sumResidualTax < 0.
Comment	

ID	G0013
Error code	INCORRECT_SUMTRANSFERABLETAX
Error message	The amount of receivables available for roll-over to the next period cannot be negative.
Node	declarationSummary/sumTransferableTax
Condition	declarationSummary/sumTransferableTax < 0.
Comment	

ID	G0014
Error code	CONFLICTING_SUMMARY_VALUES1
Error message	Conflicting summary fields (only one of the amount of tax refundable and the amount of tax payable can be greater than 0).
Node	declarationSummary
Condition	declarationSummary/sumPayableTax > 0 AND declarationSummary/sumDeductibleTax > 0
Comment	

ID	G0015
Error code	CONFLICTING_SUMMARY_VALUES2
Error message	Conflicting summary fields (if the amount of tax payable is greater than zero, the amount of receivables available for roll-over to the next period cannot be greater than zero).
Node	declarationSummary
Condition	declarationSummary/sumPayableTax > 0 AND declarationSummary/sumTransferableTax != 0.
Comment	

ID	G0017
Error code	MISSING_ADDITIONALATTACHMENTINFO
Error message	Some details of the attached factoring contract are missing.
Node	attachment
Condition	attachment/attachmentCategory == "FACTORING_CONTRACT" AND attachment/AdditionalAttachmentInfo/FactoringContractData node does not exist.
Comment	

ID	G0018
Error code	CONFLICTING_TOTALROWCOUNT_SUMPAYABLETAX
Error message	Conflicting fields (if the analytics do not include a voucher line, the amount of tax payable cannot be greater than 0).
Node	declarationSummary/sumPayableTax
Condition	Number of items in the analytics == 0 AND declarationSummary/sumPayableTax > 0.
Comment	

ID	G0019
Error code	CONFLICTING_TOTALROWCOUNT_SUMDEDUCTIBLETAX
Error message	Conflicting fields (if the analytics do not include a voucher line or a decreasing item available for setoff from the previous period, the amount of tax refundable cannot be greater than 0).
Node	declarationSummary
Condition	Number of items in the analytics == 0 AND declarationSummary/sumResidualTax is not filled out AND declarationSummary/sumDeductibleTax > 0.
Comment	

ID	G0020
Error code	TOO_MANY_ATTACHMENT
Error message	More than one attachment has been filled in!
Node	vatDeclarationData/attachment
Condition	more than 1 attachment node exists
Comment	

ID	G0021
Error code	PERIOD_INTERVAL_FREQUENCY_MISMATCH
Error message	The start and end of the declaration period does not match the declaration frequency specified!
Node	vatDeclarationData/declarationInfo/declarationFrequency
Condition	<p>The period defined by the analytics does not match the reporting frequency, i.e.:</p> <ul style="list-style-type: none"> • declarationInfo.declarationFrequency = 'MONTHLY' AND declarationInfo.declarationPeriodEnd - declarationInfo.declarationPeriodStart is not between 1 and 31 • declarationInfo.declarationFrequency = 'QUARTERLY' and declarationInfo.declarationPeriodEnd - declarationInfo.declarationPeriodStart is not between 1 and 93 • declarationInfo.declarationFrequency = 'ANNUAL' AND declarationInfo.declarationPeriodEnd - declarationInfo.declarationPeriodStart is not between 1 and 366
Comment	

ID	G0022
Error code	INVALID_FACTORINGAMOUNT_LENGTH
Error message	The number of integer places in the 'factored amount' field exceeds the maximum allowed!
Node	attachment/additionalAttachmentInfo/factoringContractData/factoringAmount
Condition	The number of places of factoringAmount without decimal places is greater than or equal to 16.
Comment	

ID	G0023
Error code	CLAIM_CHECK_ID_MISMATCH
Error message	The attachment IDs provided in the upload service do not match the attachment IDs provided in the declaration XML.
Node	attachment
Condition	The attachment IDs specified in the upload service do not match the attachment IDs specified in the XML.
Comment	

ID	G0024
Error code	INVALID_CONFLICTING_SUMACCOUNTEDTAX_SUMPAYABLETAX
Error message	The value of the field 'Amount of tax payable' provided does not match the value provided in the field 'The difference between the combined amount of tax assessed for the current period and deductible input tax'!
Node	VatDeclarationData/declarationSummary/sumPayableTax
Condition	(VatDeclarationData/declarationSummary/sumPayableTax > 0 AND VatDeclarationData/declarationSummary/sumAccountedTax < 0) OR (VatDeclarationData/declarationSummary/sumAccountedTax <> VatDeclarationData/declarationSummary/sumPayableTax)
Comment	

ID	G0025
Error code	INVALID_CONFLICTING_SUMACCOUNTEDTAX_SUMDEDUCTIBLETAX_1
Error message	The ‘Amount of tax refundable’ field cannot be filled (>0) if ‘The difference between the combined amount of tax assessed for the current period and deductible input tax’ is positive!
Node	VatDeclarationData/declarationSummary/sumDeductibleTax
Condition	VatDeclarationData/declarationSummary/sumDeductibleTax filled out AND VatDeclarationData/declarationSummary/sumAccountedTax >= 0
Comment	

ID	G0026
Error code	INVALID_CONFLICTING_SUMACCOUNTEDTAX_SUMDEDUCTIBLETAX_2
Error message	The ‘Amount of tax refundable’ field cannot be filled if the ‘Return decision’ field is not filled.
Node	VatDeclarationData/declarationSummary/sumDeductibleTax
Condition	VatDeclarationData/declarationSummary/sumDeductibleTax is filled AND VatDeclarationData/declarationStatements/returnDecision is not filled.
Comment	

ID	G0027
Error code	INVALID_CONFLICTING_SUMACCOUNTEDTAX_SUMDEDUCTIBLETAX_3
Error message	The ‘Amount of tax refundable’ field cannot be filled if ‘The difference between the combined amount of tax assessed for the current period and deductible input tax’ is greater than HUF -1,000k for a monthly payer, HUF -250k for a quarterly payer, or HUF -50k for an annual payer, or
Node	VatDeclarationData/declarationSummary/sumDeductibleTax
Condition	<p>VatDeclarationData/declarationSummary/sumDeductibleTax is filled</p> <p>AND</p> <p>VatDeclarationData/declarationSummary/sumAccountedTax > -1000 AND declarationInfo/declarationFrequency == "MONTHLY"</p> <p>OR</p> <p>VatDeclarationData/declarationSummary/sumAccountedTax > -250 AND declarationInfo/declarationFrequency == "QUARTERLY"</p> <p>OR</p> <p>VatDeclarationData/declarationSummary/sumAccountedTax > -50 AND declarationInfo/declarationFrequency == "ANNUAL".</p> <p>Values are in thousands of forints.</p>
Comment	

ID	G0028
Error code	INVALID_CONFLICTING_RETURNDECISION_SUMTRANSFERABLETAX
Error message	If ‘Return decision’ is filled, ‘Receivables available for roll-over to the next period’ cannot be filled
Node	VatDeclarationData/declarationStatements/returnDecision
Condition	VatDeclarationData/declarationStatements/returnDecision is filled AND VatDeclarationData/declarationSummary/sumTransferableTax is filled
Comment	

ID	G0029
Error code	NOT_ALLOWED_DECLARATIONPERIODSTART
Error message	The start of the declaration period cannot be earlier than 2024-01-01
Node	-
Condition	DECLARATION_INFO.PERIOD_START < 2024-01-01
Comment	

ID	G0030
Error code	CONFLICTING_PERIODSTART_PERIODEND
Error message	The start of the declaration period cannot be greater than the end of the declaration period.
Node	-
Condition	vatDeclarationData/declarationInfo/declarationPeriodStart vatDeclarationData/declarationInfo/declarationPeriodEnd >
Comment	

ID	G0031
Error code	CONFLICTING_PERIODEND_CATALOGVALIDTO
Error message	The end of the declaration period cannot be greater than the end date of the tax code catalogue validity period.
Node	-
Condition	vatDeclarationData/declarationInfo/declarationPeriodEnd cannot be greater than the end date of the tax code catalogue validity period.
Comment	

ID	G0032
Error code	CONFLICTING_MODIFICATIONREFERENCE
Error message	For basic declarations, filling the details of the amendment is not allowed.
Node	declarationInfo/modificationReference
Condition	The declaration type is basic declaration and vatDeclarationData/declarationInfo/navCorrection = false AND vatDeclarationData/declarationInfo/modificationReference node exists
Comment	

ID	G0033
Error code	MISSING_BARCODE
Error message	If correcting a declaration, it is required to indicate the identifier of the declaration to be corrected.
Node	declarationInfo/modificationReference/barCodeReference
Condition	vatDeclarationData/declarationInfo/navCorrection = true AND VatDeclarationData/declarationInfo/modificationReference/barCodeReference = null
Comment	

ID	G0034
Error code	CONFLICTING_METHOD_SUMMARY
Error message	Filling in the originalDeclarationSummary node is only allowed for self-audits!
Node	originalDeclarationSummary
Condition	The declaration type is not self-audit and originalDeclarationSummary node exists
Comment	

The following rules are run on all analytics items

ID	I0001
Error code	CONFLICTING_STANDARDTAXCODE_SEVENPERCENTCOUNT
Error message	The value of the field sevenPercentCount (7% agricultural compensation surcharge count) is incorrect for standard tax code!
Node	vatAnalytics
Condition	vatAnalytics/vatAnalyticsItem/taxInformation/standardTaxCode field contains the code "MD72" AND vatAnalytics/agriculturalCompensationPremium/sevenPercentCount does not exist.
Comment	

ID	I0002
Error code	CONFLICTING_STANDARDTAXCODE_TWELVEPERCENTCOUNT
Error message	The value of the field twelvePercentCount (12% agricultural compensation surcharge count) is incorrect for standard tax code!
Node	vatAnalytics
Condition	vatAnalytics/vatAnalyticsItem/taxInformation/standardTaxCode field contains the code "MD73" AND vatAnalytics/AgriculturalCompensationPremium/twelvePercentCount does not exist.
Comment	

ID	I0003
Error code	LINE_NUMBER_NOT_SEQUENTIAL
Error message	The analytics items are non-continuous or duplicated!
Node	vatAnalytics/VatAnalyticsItem
Condition	Checking for sequence continuity and duplicate indexes.
Comment	

ID	I0004
Error code	DUPLICATED_SOURCEDOCUMENTID
Error message	Illegal duplication between the unique IDs of source documents found (the same voucher ID is used more than once for a domestic partner)!
Node	vatAnalytics/VatAnalyticsItem/sourceDocumentId
Condition	sourceDocumentId uniqueness verification (verification per partner if vatAnalytics/VatAnalyticsItem/PartnerInfo/partnerStatus = "DOMESTIC").
Comment	

ID	I0005
Error code	INVALID_SOURCEDOCUMENTISSUEDATE
Error message	The issue date of the source document cannot be a future date
Node	vatAnalytics/VatAnalyticsItem/sourceDocumentIssueDate
Condition	vatAnalytics/VatAnalyticsItem/sourceDocumentIssueDate > sysdate.
Comment	

ID	I0006
Error code	INVALID_TAXPOINTDATE
Error message	The tax point (taxpointDate) cannot be a future date!
Node	vatAnalytics/VatAnalyticsItem/taxpointDate
Condition	vatAnalytics/VatAnalyticsItem/taxpointDate > sysdate.
Comment	

ID	I0008
Error code	MISSING_PARTNER_INFO
Error message	Incomplete partner data for partner VAT status field value!
Node	vatAnalytics/VatAnalyticsItem/PartnerInfo
Condition	vatAnalytics/VatAnalyticsItem/PartnerInfo/partnerStatus ("DOMESTIC" OR "OTHER") AND (vatAnalytics/VatAnalyticsItem/PartnerInfo/partnerTaxData OR vatAnalytics/VatAnalyticsItem/PartnerInfo/partnerName OR vatAnalytics/VatAnalyticsItem/PartnerInfo/partnerAddress) is not specified.
Comment	

ID	I0009
Error code	MISSING_DOMESTICTAXDATA
Error message	Despite domestic VAT taxpayer status, the domestic tax number is missing!
Node	vatAnalytics/VatAnalyticsItem/PartnerInfo/
Condition	vatAnalytics/VatAnalyticsItem/PartnerInfo/partnerStatus == "DOMESTIC" AND vatAnalytics/VatAnalyticsItem/PartnerInfo/PartnerTaxData/domesticTaxData is not specified.
Comment	

ID	I0011
Error code	INVALID_TAXCODE
Error message	Non-existent tax code!
Node	vatAnalytics/VatAnalyticsItem/TaxInformation/standardTaxCode
Condition	<p>vatAnalytics/VatAnalyticsItem/TaxInformation/standardTaxCode not listed in the tax code catalogue</p> <p>Alternately, it is in the catalogue, but the value of vatAnalytics/VatAnalyticsItem/taxpointDate is not within the validity period (valid_from – valid_to) of the tax code</p>
Comment	

ID	I0012
Error code	INVALID_TAXPOSITION_COUNT
Error message	The item tax position field is provided more than once!
Node	vatAnalytics/VatAnalyticsItem/TaxInformation/TaxPosition/positionType
Condition	under the vatAnalytics/VatAnalyticsItem/TaxInformation/ node, the taxPosition node is listed twice AND the positionType value has the value 'PAYABLE' or 'DEDUCTBLE' twice.
Comment	

ID	I0013
Error code	INVALID_TAXBASE
Error message	The absolute value of the tax base cannot be less than the absolute value of the tax amount!
Node	vatAnalytics/VatAnalyticsItem/TaxInformation/TaxPosition
Condition	ABS(vatAnalytics/VatAnalyticsItem/TaxInformation/TaxPosition/taxBase) < ABS(vatAnalytics/VatAnalyticsItem/TaxInformation/TaxPosition/taxAmount.)
Comment	

ID	I0014
Error code	INVALID_DEDUCTION_OPTIONS
Error message	The VAT position indication does not match to the filling of deduction data!
Node	vatAnalytics/VatAnalyticsItem/TaxInformation/TaxPosition
Condition	vatAnalytics/VatAnalyticsItem/TaxInformation/TaxPosition/positionType == "PAYABLE" vatAnalytics/VatAnalyticsItem/TaxInformation/TaxPosition/DeductionOptions exists. AND node
Comment	

ID	I0015
Error code	INVALID_DEDUCTION_AMOUNT
Error message	The tax amount of the item cannot be less than the deduction amount!
Node	vatAnalytics/VatAnalyticsItem/TaxInformation/TaxPosition/deductionAmount
Condition	VatDeclarationData/VatAnalytics/VatAnalyticsItem/TaxInformation/TaxPosition/taxAmount < VatDeclarationData/VatAnalytics/VatAnalyticsItem/TaxInformation/TaxPosition/DeductionOptions/deductionAmount.
Comment	

ID	I0016
Error code	INVALID_TAXBASE_LENGTH
Error message	The length of the Tax base field cannot exceed 13 integer places!
Node	vatAnalytics/VatAnalyticsItem/TaxInformation/TaxPosition/taxBase
Condition	vatAnalytics/VatAnalyticsItem/TaxInformation/TaxPosition/taxBase is longer than 13 integer places.
Comment	

ID	I0017
Error code	INVALID_TAXAMOUNT_LENGTH
Error message	The length of the Tax amount field cannot exceed 13 integer places!
Node	vatAnalytics/VatAnalyticsItem/TaxInformation/TaxPosition/taxAmount
Condition	vatAnalytics/VatAnalyticsItem/TaxInformation/TaxPosition/taxAmount is longer than 13 integer places.
Comment	

ID	I0018
Error code	INVALID_DEDUCTIONAMOUNT_LENGTH
Error message	The length of the Deduction amount field cannot exceed 13 integer places!
Node	vatAnalytics/VatAnalyticsItem/TaxInformation/TaxPosition/DeductionOptions/deductionAmount
Condition	vatAnalytics/VatAnalyticsItem/TaxInformation/TaxPosition/DeductionOptions/deductionAmount is longer than 13 integer places.
Comment	

ID	I0019
Error code	STANDARDTAXCODE_PAYABLE_NOT_ALLOWED
Error message	The standard tax code provided cannot be included in the deduction with a payable tax position!
Node	vatAnalytics/VatAnalyticsItem/TaxInformation/standardTaxCode
Condition	the value vatAnalytics/VatAnalyticsItem/TaxInformation/taxPosition/positionType = 'PAYABLE' is specified for vatAnalytics/VatAnalyticsItem/TaxInformation/standardTaxCode, but the value of the payable field associated to it in the tax code catalogue in force is false
Comment	

ID	I0020
Error code	STANDARDTAXCODE_DEDUCTIBLE_NOT_ALLOWED
Error message	The standard tax code provided cannot be included in the deduction with a deductible tax position!
Node	vatAnalytics/VatAnalyticsItem/TaxInformation/standardTaxCode
Condition	the value vatAnalytics/VatAnalyticsItem/TaxInformation/taxPosition/positionType = 'DEDUCTIBLE' is specified for vatAnalytics/VatAnalyticsItem/TaxInformation/standardTaxCode, but the value of the deductible field associated to it in the tax code catalogue in force is false
Comment	

ID	I0021
Error code	STANDARDTAXCODE_OTHER_NOT_ALLOWED
Error message	The standard tax code provided cannot be included in the deduction with the 'OTHER' tax position!
Node	vatAnalytics/VatAnalyticsItem/TaxInformation/standardTaxCode
Condition	the value vatAnalytics/VatAnalyticsItem/TaxInformation/taxPosition/positionType = 'OTHER' is specified for vatAnalytics/VatAnalyticsItem/TaxInformation/standardTaxCode, but the value of the deductible or payable field associated to it in the tax code catalogue in force is true
Comment	

The following rules are evaluated on sheetList nodes

ID	S0003
Error code	DUPLICATED_SHEETNAME
Error message	Illegal duplication between the names of supplementary sheets found!
Node	sheetList/sheet/sheetName
Condition	Duplicate sheetName found in any sheetList/sheet/sheetName field
Comment	

ID	S0005
Error code	SHEET_PAGE_COUNT_EXCEEDED
Error message	The number of supplementary sheets has exceeded the maximum value allowed!
Node	sheetList.sheet.sheetPageCount
Condition	<p>sheetList/sheet/sheetName = VAT_SHEET_170 és sheetList/sheet/sheetPageCount értéke !=1 VAGY</p> <p>sheetList/sheet/sheetName = VAT_SHEET_4 és sheetList/sheet/sheetPageCount értéke >9 VAGY</p> <p>sheetList/sheet/sheetName in ('VAT_SHEET_2','VAT_SHEET_EUNY') és sheetList/sheet/sheetPageCount értéke >99 VAGY</p> <p>sheetList/sheet/sheetName in ('VAT_SHEET_3','VAT_SHEET_6','VAT_SHEET_7','VAT_SHEET_8','VAT_SHEET_9','VAT_SHEET_A88') és sheetList/sheet/sheetPageCount értéke >999)</p>
Comment	

ID	S0006
Error code	NOT_ALLOWED_SHEETLIST_NAVFIELDID
Error message	Not allowed navFieldId on the supplementary sheet!
Node	sheetList
Condition	At least one of the sheetList/sheet/sheetField/sheetFieldName values does not match the structure of the allowed navFieldId values for the sheetList/sheet/sheetName supplementary sheet.
Comment	-

ID	S0007
Error code	DUPLICATED_SHEETLIST_NAVFIELDID
Error message	The navFieldId in the sheetFieldName was duplicated in the supplementary sheet!
Node	sheetList
Condition	At least one of the sheetList/sheet/sheetField/sheetFieldName values are duplicated in the XML
Comment	-

ID	S0008
Error code	CONFLICTING_SHEET_VAT_SHEET_4
Error message	When submitting a basic declaration or corrigendum, the submission of supplementary sheet 4 is not allowed.
Node	sheetList/sheet/sheetName
Condition	The declaration type is basic declaration or corrigendum AND if 'VAT_SHEET_4' sheetName can be found in any sheetList/sheet/sheetName field.
Comment	-

ID	S0009
Error code	CONFLICTING_SHEET_VAT_SHEET_EUNY
Error message	When submitting a basic declaration, corrigendum or correction, submission of the supplementary sheet EUNY is not allowed!
Node	sheetList/sheet/sheetName
Condition	(The declaration type is not self-audit OR the declaration type is self-audit and VatDeclarationData\declarationInfo\navCorrection= 1 (true)) 'VAT_SHEET_EUNY' sheetName can be found in any sheetList/sheet/sheetName field.
Comment	-

ID	S0010
Error code	MISSING_04_SHEETLIST
Error message	When submitting a self-audit, supplementary sheet 4 must be filled out.
Node	VatDeclarationData/SheetList/sheet/sheetName
Condition	The declaration type is self-audit and 'VAT_SHEET_4' is not found in any sheetList/sheet/sheetName field
Comment	-

ID	S0011
Error code	MISSING_INDIRECT_CUSTOMS_SHEET
Error message	Missing Annex 3, with data appearing in the corresponding summary fields
Intersection	vatDeclarationData/sheetList/sheet/sheetName
Requirement	DeclarationAdditionalData/ImportedTaxDeductAssessed <> 0 AND DeclarationAdditionalData/ImportedTaxDeductSelfAssess <> 0 AND a sheetList/sheet/sheetName nem tartalmazza 'VAT_SHEET_3'
Note	-

ID	S0012
Error code	MISSING_INDIRECT_CUSTOMS_DATA
Error message	Hiányoznak a 3-as melléklaphoz tartozó összesítő adatok. The aggregated data for Appendix 3 are missing.
Intersection	vatDeclarationData/sheetList/sheet/sheetName
Requirement	A sheetList/sheet/sheetName contains the 'VAT_SHEET_3' value And DeclarationAdditionalData/ImportedTaxDeductAssessed is null AND DeclarationAdditionalData/ImportedTaxDeductSelfAssess is null
Note	-

The following rules examine the whole of the analytics in context

ID	F0001
Error code	CONFLICTING_TOTALROWCOUNT_ANALYTICSITEMCOUNT
Error message	The value of the 'number of lines in the analytics' field does not match the number of lines in the analytics!
Node	VatDeclarationData/VatAnalytics/totalRowCount
Condition	VatDeclarationData/VatAnalytics/totalRowCount != the number of rows calculated during processing
Comment	

ID	F0004
Error code	MISSING_SHEETLIST
Error message	The corresponding supplementary sheet cannot be found for the standard tax code provided!
Node	VatDeclarationData/VatAnalytics/VatAnalyticsItem/taxInformation/standardTaxCode
Condition	The mandatory supplementary sheet to the tax code provided has not been filled out.
Comment	

ID	F0005
Error code	MISSING_STANDARD_TAX_CODE
Error message	The corresponding standard tax code cannot be found for the supplementary sheet provided!
Node	sheetList/sheet/sheetName
Condition	No relevant tax code for VatDeclarationData/sheetList/sheet/sheetName found for supplementary sheets 2, 6, 7, 8, 9, A88.
Comment	

ID	F0006
Error code	DUPLICATED_SOURCEDOCUMENTID
Error message	Illegal duplication between the unique IDs of source documents found (the same voucher ID is used more than once for a domestic partner)!
Node	VatDeclarationData/VatAnalytics/VatAnalyticsItem/sourceDocumentId
Condition	multiple references to sourceDocumentId (verification per partner if vatAnalytics/VatAnalyticsItem/PartnerInfo/partnerStatus = "DOMESTIC").
Comment	

ID	F0007
Error code	INVALID_CONFLICTING_SUMTAXAMOUNT_SUMACCOUNTEDTAX
Error message	The value of the field ‘The difference between the combined amount of tax assessed for the current period and deductible input tax’ provided does not match the difference between the amount of tax payable and deductible in the analytics and ‘The amount of decreasing item available for setoff from the previous period’ with the correct sign!
Node	VatDeclarationData/declarationSummary/sumAccountedTax
Condition	VatDeclarationData/declarationSummary/sumAccountedTax != ‘The difference between the combined amount of tax assessed for the current period and deductible input tax’ (total tax payable - total tax deductible - declarationSummary/sumResidualTax) calculated during processing
Comment	

ID	F0008
Error code	NOT_ALLOWED_DEDUCTION_CUSTOMS_TAX_DATA
Error message	It is not permitted to fill in Sheet 3 and related data for the period specified.
Intersection	VatDeclarationData/declarationInfo/declarationPeriodEnd
Requirement	<p>VatDeclarationData/declarationInfo/declarationPeriodEnd $< '2025-03-01'$</p> <p>AND</p> <p>DeclarationAdditionalData/ImportedTaxDeductAssessed $\neq 0$</p> <p>OR DeclarationAdditionalData/ImportedTaxDeductSelfAssess $\neq 0$</p> <p>OR tartalmazza a sheetList/sheet/sheetName = ‘VAT_SHEET_3’</p>
Megjegyzés	

WARNING type messages

ID	E0001
Error code	INVOICE_TAXPOINTDATE_LATE
Error message	The end of the declaration period is greater than 31 December of the year following the year of delivery!
Node	vatAnalytics/vatAnalyticsItem/taxpointDate
Condition	vatAnalytics/vatAnalyticsItem/taxInformation/taxPosition/positionType = "DEDUCTIBLE" AND DECLARATION_INFO.PERIOD_END is greater than 31 December of the year following the year of taxPointDate
Comment	

ID	E0002
Error code	SUPPLIER_CUSTOMER_MATCH_TAXPAYER
Error message	Seller and buyer cannot have the same tax number!
Node	vatAnalytics/vatAnalyticsItem/partnerInfo/partnerTaxData/domesticTaxData/taxNumber
Condition	vatAnalytics/vatAnalyticsItem/sourceDocumentType = 'INVOICE' AND the tax number of the submitter of the analytics is the same as the one in vatAnalytics/vatAnalyticsItem/partnerInfo/partnerTaxData/domesticTaxData/taxNumber AND the value at the 9th place of the tax number is not 4 or 5
Comment	

ID	E0004
Error code	TAXPOINTDATE_NOT_IN_DECLARATION_PERIOD
Error message	The tax date value for the voucher does not fall within the declaration period (in the case of a voucher payable). May be explained by cash settlement.
Node	vatAnalytics/VatAnalyticsItem/taxpointDate
Condition	vatAnalytics/VatAnalyticsItem/TaxInformation/TaxPosition/positionType = 'PAYABLE' AND vatAnalytics/VatAnalyticsItem/taxpointDate does not fall between declarationInfo/declarationPeriodStart and declarationInfo/declarationPeriodEnd.
Comment	

ID	E0005
Error code	TAXPOINTDATE_TOO_EARLY
Error message	The outbound invoice in the analytics was issued more than 10 days earlier than the current period.
Node	vatAnalytics/VatAnalyticsItem/taxpointDate
Condition	vatAnalytics/VatAnalyticsItem/sourceDocumentType = 'INVOICE' AND vatAnalytics/VatAnalyticsItem/TaxInformation/TaxPosition/positionType = 'PAYABLE' AND vatAnalytics/VatAnalyticsItem/taxpointDate < declarationInfo/declarationPeriodStart -10 days
Comment	

ID	E0006
Error code	TAXPOINTDATE_LATE
Error message	The tax date (delivery date) of the voucher deducted is the date following the reference period (declaration period).
Node	vatAnalytics/VatAnalyticsItem/taxpointDate
Condition	vatAnalytics/VatAnalyticsItem/TaxInformation/TaxPosition/positionType = 'DEDUCTIBLE' AND vatAnalytics/VatAnalyticsItem/taxpointDate > declarationInfo/declarationPeriodEnd
Comment	

ID	E0007
Error code	TOTAL_ROW_COUNT_REGISTER_DATA_MISMATCH
Error message	You have submitted a null VAT analytics, but issued online invoice and/or OPG and/or CDPS information exist for the declaration period.
Node	vatAnalytics/totalRowCount
Condition	vatAnalytics/totalRowCount = 0 AND issued online invoice and/or OPG and/or CDPS information exist for the taxpayer.
Comment	

ID	E0008
Error code	INCOMPATIBLE_TAXCODES_1
Error message	Domestic sales tax codes are incompatible with community sales tax codes if associated to the same document ID!
Node	vatAnalytics/VatAnalyticsItem/taxinformation/standardTaxCode
Condition	Among the tax codes vatAnalyticsItem/taxinformation/standardTaxCode within an analytics item (vatAnalyticsItem), tax codes starting with MP02 or MP03 can be found next to ones starting with MP04, MP05, MP06, MP07, MP08, MP09.
Comment	If there are multiple conflicting codes, only the first one is included in the tag.

ID	E0009
Error code	INCOMPATIBLE_TAXCODES_2
Error message	Domestic sales tax codes are incompatible with sales tax codes from a third country if associated to the same document ID!
Node	vatAnalytics/VatAnalyticsItem/taxinformation/standardTaxCode
Condition	Among the tax codes vatAnalyticsItem/taxinformation/standardTaxCode within an analytics item (vatAnalyticsItem), tax codes starting with MP01 can be found next to ones starting with MP04, MP05, MP06, MP07, MP08, MP09.
Comment	If multiple codes conflict, only the first one is put in the tag

ID	E0010
Error code	INCOMPATIBLE_TAXCODES_3
Error message	Domestic purchase tax codes are incompatible with community purchase tax codes if associated to the same document ID!
Node	vatAnalytics/VatAnalyticsItem/taxinformation/standardTaxCode
Condition	Among the tax codes vatAnalyticsItem/taxinformation/standardTaxCode within an analytics item (vatAnalyticsItem), tax codes starting with MD67 or MD69 can be found next to ones starting with MD63, MD64, MD65, MD66, MD68, MD72, MD73, MD74, MD75.
Comment	If multiple codes conflict, only the first one is put in the tag

ID	E0011
Error code	INCOMPATIBLE_TAXCODES_4
Error message	Domestic purchase tax codes are incompatible with purchase tax codes from a third country if associated to the same document ID!
Node	vatAnalytics/VatAnalyticsItem/taxinformation/standardTaxCode
Condition	Among the tax codes vatAnalyticsItem/taxinformation/standardTaxCode within an analytics item (vatAnalyticsItem), tax codes starting with MD67, MD70 or MD71 can be found next to ones starting with MD63, MD64, MD65, MD66, MD68, MD72, MD73, MD74, MD75.
Comment	If multiple codes conflict, only the first one is put in the tag

ID	E0012
Error code	INCOMPATIBLE_TAXCODES_5
Error message	The tax codes for domestic purchases cannot be provided on an invoice where the VAT status of the partner is 'OTHER'!
Node	vatAnalytics/VatAnalyticsItem/taxinformation/standardTaxCode
Condition	The value vatAnalyticsItem/partnerInfo/partnerStatus = 'OTHER' is provided on an analytics item AND the standard tax codes of the analytics item (vatAnalyticsItem/taxinformation/standardTaxCode) include one starting with MD63 or MD64 or MD65 or MD66 or MD68 or MD72 or MD73 or MD74 or MD75.
Comment	If multiple codes conflict, only the first one is put in the tag

ID	E0013
Error code	INCOMPATIBLE_TAXCODES_6
Error message	The tax codes for community sales, sales to third countries, community purchases and purchases from third countries cannot be provided on an invoice where the VAT status of the partner is 'Domestic VAT taxpayer'!
Node	vatAnalytics/VatAnalyticsItem/taxinformation/standardTaxCode
Condition	The value vatAnalyticsItem/partnerInfo/partnerStatus = 'DOMESTIC' is provided on an analytics item AND the tax codes of the analytics item (vatAnalyticsItem/taxinformation/standardTaxCode) include one starting with MP02 or MP03 or MP01 or MD67 or MD69 or MD70 or MD71.
Comment	If multiple codes conflict, only the first one is put in the tag

ID	E0014
Error code	CONFLICTING_INCREASE_ALLOWANCE
Error message	There was an increase in payment obligations, but no self-revision surcharge was calculated.
Node	-
Condition	<p>WARN must be registered if:</p> <p>the declaration type is self-audit and there was an increase in payment obligations (basic declaration Amount of tax payable - self-audit Amount of tax payable > 0), however, the self-audit surcharge field was not filled out.</p>
Comment	

ID	E0015
Error code	CONFICTING_REDUCTION_ALLOWANCE
Error message	There was a reduction in payment obligations, but a self-revision surcharge was calculated.
Node	-
Condition	<p>WARN must be registered if:</p> <p>the declaration type is self-audit and</p> <p>there was a reduction in payment obligations (basic declaration Amount of tax payable - self-audit Amount of tax payable < 0), however, the self-audit surcharge field was filled out.</p>
Comment	

ID	E0016
Error code	NO_ALLOWANCE_REQUIRED
Error message	The basic return submitted using the eVAT M2M interface was self-audited within 15 days of the due date, but a self-audit surcharge was charged.
Node	-
Condition	<p>WARN must be registered if:</p> <p>the declaration type is self-audit, and</p> <p>the analytics were submitted within 15 days of the end of the declaration period and the self-audit surcharge field was filled out.</p>
Comment	

ID	E0017
Error code	TAX_CODE_NOT_SELF_CHECK
Error message	In cases regulated by Sections 153/B and 153/C of the VAT Act, the tax settlement is not subject to a self-audit, but must instead be settled in the next declaration. (Tax code MP31)
Node	-
Condition	<p>WARN must be registered if:</p> <p>the declaration type is self-audit, and</p> <p>there is at least one analytical element for which the tax code does not require the analytical element to be submitted as a self-audit.</p>
Comment	

ID	PE0001
Error code	CONFLICTING_STANDARDTAXCODE
Error message	The submitted declaration includes an analytical item that is subject to VAT, but is listed as VAT exempt in the Online Invoice System.
Node	vatDeclarationData/vatAnalytics/vatAnalyticsItem/sourceDocumentId
Condition	A WARN must be registered if the tax code(s) of the invoice in the XML submitted by the user does not match the tax code(s) of the invoice submitted by the user for the online invoice.
Comment	

ID	PE0002
Error code	CONFLICTING_SOURCEDOCUMENTID
Error message	The submitted declaration includes a deducted analytical item that is not listed in the Online Invoice System.
Node	vatDeclarationData/vatAnalytics/vatAnalyticsItem/sourceDocumentId
Condition	A WARN must be registered if a deduction invoice in the XML submitted by the user is not included in the taxpayer's online invoices.
Comment	

ID	PE0003
Error code	CONFLICTING_POSITIONTYPE
Error message	A deductible item in the submitted declaration is shown as payable in the Online Invoice System.
Node	vatDeclarationData/vatanalitics/vatAnalyticalItem /taxinformation/taxPosition/positionType
Condition	A WARN must be registered if a deduction invoice in the XML submitted by the taxpayer is listed as payable in the taxpayer's online invoices.
Comment	

ID	PEF0001
Error code	CONFLICTING_SUM_TAXAMOUNT
Error message	The total amount of deductibles in the submitted declaration does not match the total amount of deductibles in the Online Invoice System.
Node	
Condition	A WARN must be registered if the deductible amount in the Online Invoice Data Report is less than what is shown in the analytics.
Comment	

INFO type messages

ID	E0003
Error code	STANDARD_TAX_CODE_POSITION_TYPE_MISMATCH
Error message	Only sales exempt from VAT are included in the analytics, but a deductible item is also provided! (with the exception of the case under Section 121 (c) of the VAT Act)
Node	
Condition	If the items submitted only include the tax code exempt from VAT (TAM), the analytics should not include a deductible item.
Comment	