

EKAER Management Service

Contents

1	Introduction	3
1.1	Objective	3
1.2	Uploading XML on the EKAER WEB interface	3
2	Structure of operations, display in XML structure	4
2.1	Basic elements defined in XSD	4
2.2	General structure of XML elements	4
2.2.1	Header XML part	4
2.2.2	User XML part	5
2.2.3	Generating a requestSignature	6
2.3	ManageTradeCardsRequest, (create, modify, delete)	7
2.3.1	TradeCardOperations	7
2.3.1.1	Create operation	8
2.3.1.2	Modify operation	8
2.3.1.3	Delete operation	8
2.3.1.4	Finalize operation	8
2.3.2	Structure of the TradeCard element	9
2.3.2.1	TradeCard data	9
2.3.2.2	Verification of the data given in the operation	11
2.3.2.3	Structure and fields of the load and unload address data element	12
2.3.2.4	Address data verification	12
2.3.2.5	Country list	13
2.3.2.6	Items list structure (tradeCardItem)	14
2.3.2.7	Verifications related to items	15
2.3.2.8	tradeReason	15
2.4	ManageTradeCardsResponse, Structure of the response	16
2.4.1	OperationResult result	16
2.4.1.1	Result structure (OperationResultType)	17
2.4.1.2	tradeCardInfo element structure	17

2.4.1.3	Operation statuses (status).....	20
2.5	Process and steps of assurance calculation.....	21
2.6	QueryTradeCards, query operations	21
3	Technical description of the service	21
3.1	General technical data	21
3.2	Operations	22
3.3	HTTP Headers.....	22
3.4	HTTP Status codes.....	22
3.5	Result element in the response message	23
3.5.1	ReasonCode Enumerated types.....	23
4	Annex	24
4.1	Example XMLs	25
4.2	Accessibility of the test system.....	25

Version

Name	Date	Version	Change briefly
B. G.	10.12.2014	1.0	initial
K. B.	12.12.2014	1.1	proofread
B. G.	13.12.2014	1.2	validation supplementation, reasonCodes, country list, Test system access

1 INTRODUCTION

The Electronic Trade and Transport Control System (*Elektronikus Kereskedelmi és Áruforgalom Ellenőrző Rendszer* (hereinafter referred to as: EKAER)) is to be launched, in which trade activities, transportation and movement of goods have to be registered in cases and in the manner stipulated by the law in the course of trade activities (hereinafter referred to as: operation). Operations of trade activities in the EKAER can be managed in the following manner:

- On the GUI of the WEB interface
- On the WEB via XML file upload
- Through services supporting machine to machine communication

The specification was made in accordance with the conditions and legal regulations known at the moment of preparation. In case regulatory or legal requirements change, the specification will change as well.

Difference between the WEB interface and the XML-based operations:

When the operation is generated on the web, it is given the status „Under planning” and no automatic EKAER number is given, while **in case of XML-based communication the operation is automatically in active status, and given an EKAER number**, and the necessary assurance calculation is also performed.

1.1 OBJECTIVE

The objective of the present document is to provide information about the XML structure used by the service supporting XML file upload and machine to machine communication and the use of the service describing machine to machine communication.

The XML file structure uploaded on the WEB and returned in response and the XML structure used by the service are identical. The same file has to be uploaded on the web as the XML structure necessary to address the service.

1.2 UPLOADING XML ON THE EKAER WEB INTERFACE

The XML structure and operations detailed in the present document can be uploaded on the WEB interface after logging in, besides direct machine to machine communication.

On the EKAER WEB interface there is a dedicated function to upload the XML file, and as it is activated, and XML response file is downloaded. The defined response in the documentation of the downloaded file will be XML.

2 STRUCTURE OF OPERATIONS, DISPLAY IN XML STRUCTURE

In this chapter the structure of XML and operations, internal logical connections related to operations and data contents are presented.

2.1 BASIC ELEMENTS DEFINED IN XSD

In the attached XSD the following elements are defined:

- **manageTradeCardsRequest:** This element is to modify, create and delete operations. In this structure, operations related to operations are transmitted in list form. Accordingly, an XML has to be uploaded on the WEB or transfer it to the service, and as a consequence the EKAER system performs the operations requested in the element.
- **manageTradeCardsResponse:** Describes the response element generated when processing the **manageTradeCardsRequest**. The EKAER system gives an XML of this structure as a response to the **manageTradeCardsRequest**.
- **queryTradeCardsRequest:** Describes the structure of the XML to query earlier registered operations. The element contains the parameters of the query.
- **queryTradeCardsResponse:** Defines the XML structure of the response given to the **queryTradeCardsRequest**. Contains the result of the query. Contains the operations matching the query as a list.

2.2 GENERAL STRUCTURE OF XML ELEMENTS

Each element has a **Header** and a **User** part. These can be found generally in every element. They contain technical fields and fields for identification.

2.2.1 HEADER XML PART

In the Header there are general technical data related to the exchange of messages. By these, the requests can be identified, requests and replies can be matched, and they also contain generally necessary technical fields.

Field name	Type	Compulsory	Description	Example
requestId	50 long text.	Yes	The unique ID of the message. Each message is to be given an ID.	1EM9C109707208L
timestamp	dateTime according to the xsd standard	Yes	Time of the creation of the request. In machine to machine communication matches the time of the request.	2014-12-05T17:10:00+01:00

requestVersion	Max 6 long text, with a Default: 1.0 value. Mask: ##.###. whole numbers separated by full stop	No, default 1.0	Contains the version number of the request. Can be used at the change of request business structure later.	1.0
headerVersion	Max 6 long text, with a Default: 1.0 value. Mask: ##.###. whole numbers separated by full stop	No, default 1.0	Contains the version number of the request. Can be used at the change of header structure later.	1.0

Limitations related to the fields:

- **The requestId has to be unique for each user.** The system does not accept a request with the same ID from different users.
- **The server does not accept requests with a timestamp older than 24 h, or a future time. There is a five minute tolerance compared to server time.**

2.2.2 USER XML PART

The User identifies the sender, the user requesting the change. In this part we can find necessary data for identification and verification.

N.B.: In case of XML upload via WEB, the modifications of operations are not performed in the name of the person uploading the XML file currently logged in, but in the name of the user identified through the data given in the User part of the XML.

Field name	Type	Compulsory	Description	Example
User	30 long text.	Yes	The login name of the user requesting the change. Login name	testelek
passwordHash	128 long text	Yes	The SHA-512 hash value of the password of the user requesting the modification. NOT THE UNCODED PASSWORD,	BA3253876AED6BC22D4A 6FF53D8406C6AD864195E D144AB5C87621B6C233B 548BAEAE6956DF346EC8C 17F5EA10F35EE3CBC5147 97ED7DDD3145464E2A0B AB413
VATNumber	8 long VAT number	Yes	The tax number of the tax subject whose operations the user intends to manage. The first 8 digits of the whole tax number.	32165498

requestSignature	128 long text	Yes	The signature of the message by which the server verifies if the user submitted the XML. A generated SHA-512 hash message based on the data in the message and the value of the secret data of the user (not in the message, but known to the system).	CE3687D87EDEFD4EAE471 BEF11C2856257B2B0CE87 9DCCB1A38049D1ABB335 CBDA49174EA4F8C8E95A AA8D7683E0734994EFA72 528E2C7EF24CC9F3B80C0 2F97
------------------	---------------	-----	--	--

The username, the password and the VAT number are the same data by which the users log in on the web interface.

2.2.3 GENERATING A REQUESTSIGNATURE

The requestSignature field prevents unauthorized users from creating operations. The hash value is verified by the server side in case of each request, and the operation is performed only if it can be actually generated based on the request received. The uniqueness of the requestId of requests is verified by the system (a given user can use a requestId only once), which is the basis of the signing hash value, therefore no new request can be created by copying the header of the request, because the hash value of the verifying requestSignature hash will not match.

The SHA-512 hash value of the text received from the connection of the text values transmitted in the field:

- requestId
- timestamp field in the following format: yyyyMMddHHmmss. e.g.: 2014.10.05 12:58:08 format: 20141005125808
- The secret signature key of the user. This password-like data can be given by the user for themselves on the WEB. It is a secret password of 8 long, which has to contain uppercase and lowercase letters and a number or numbers, e.g. titkos7Password98. XML interfaces can only be used by persons having a signature key.

Example:

The secret signature key of testekek used in the example (set by the user on the WEB interface): Elek65Titkos

Data of the example request:

- requestId = TSTKFT1222564
- timestamp = 2015.01.15T12:25:45+01, of which the value used for the hash is: 20150115122545

The text value of which the hash is created is structured like this:

TSTKFT1222564 + 20150115122545 + Elek65Titkos= TSTKFT122256420150115122545Elek65Titkos

The text generated like this („TSTKFT122256420150115122545Elek65Titkos”) would have an SHA-512 hash value like this:

AF84DC456B82234E67550C80169E517FBDAB4403607293985DECB09F534D9F73FADAABEFEE932554FA
BBC49F6E8F74A5DD54EA359D6B7644D95CFF3530AFB889

Verification can be performed on the following site:

<http://www.convertstring.com/hu/Hash/SHA512>

2.3 MANAGETRADECARDSREQUEST, (CREATE, MODIFY, DELETE)

The general part of the message (Header and User) is detailed in [2.2](#).

In the XML structure the business data are in the tradeCardOperations list.

2.3.1 TRADECARDOPERATIONS

The tradeCardOperations element contains a tradeCardOperation list, with the operations to be performed. Registering operations, modifying and deleting existing operations. The operation to be performed is described by the tradeCardOperation element.

Structure of the tradeCardOperation:

Field name	Type	Compulsory	Description	Example
index	xsd integer	Yes	An ordinal number according to the place in the list. Identifies the given modification operation within the request	1
operation	enumerated: create, modify, delete, finalize	Yes	Indicates the manner of modification. The type of the given modification task.	create
tradeCard / tcn	selection: tradeCard element or tcn element	Yes	When operation=create and operation=modify a tradeCard element is necessary. When operation=delete a tcn is necessary	

Whether there is a tradeCard or a tcn element in the given tradeCardOperation depends on the operation field.

The Operation defines what operation is performed by the server.

2.3.1.1 CREATE OPERATION

In case of “Create” operation, the tradeCardOperation has to contain a tradeCard element.

The tradeCard element contains the data of the operation, based on which the server creates the operation.

In case of creation, the tcn element within the tradeCard has to be omitted.

Within the tradeCardItem element, the id attribute has to be omitted.

2.3.1.2 MODIFY OPERATION

In case of “modify” operation, the tradeCardOperation has to contain the tradeCard element.

The tradeCard element contains data of the operation, based on which the server modifies the operation.

Logic of the modification:

Data in the header of the operation are saved.

The manner of processing the item data is the following:

- Based on the item id attribute, the server identifies the item and modifies it based on the data received. In case it is not retrieved, the modification of the whole operation will be rendered unsuccessful, the system will not perform it.
- When the request does not contain existing data, the server side deletes the given item. Therefore in modification requests non-existing items will be deleted from the items belonging the operation

No new item can be added to the already registered operation. The system does not allow that.

2.3.1.3 DELETE OPERATION

In case of Delete, only the tcn (EKAER number) has to be submitted, and the tradeCard whole object does not have to be built up. The server identifies and deletes the operation based on the EKAER number given in the tcn. The deletion can only be performed when the operation is still “active”.

2.3.1.4 FINALIZE OPERATION

In case of Finalize, only the tcn has to be submitted, and the tradeCard whole object does not have to be built up. The server identifies and finalizes the operation based on the EKAER number given in the tcn. During finalization, verifications are also performed. About these, further information is available in the following chapter: [Verification of the data given in the operation](#)

N.B.: Before finalizing the operation by Finalize, all the necessary values have to be set by Modify operation, as after finalization the system does not allow the modification of the data. E.g.. setting the data of the time of unload might be necessary to be modified before finalization.

2.3.2 STRUCTURE OF THE TRADECARD ELEMENT

In the tradeCard element data related to the operation are stored. It can be divided into two main parts: header and item list. The header contains data related to the operation, while the item list contains itemized data for the products belonging to the operation.

2.3.2.1 TRADECARD DATA

Data in the tradeCard describe the details of the operation.

Field name	Type	Compulsory	Description	Example
tcn	20 long text	compulsory in case of modify operation, otherwise can be omitted	EKAER number of the operation. Identifies the operation.	12312312331
orderNumber	50 long text	no	Identifies the operation/order in the own system of the user	ASDF234ffas3
tradeType	Enumerated: E, I, D	yes	This field defines the relation of the product movement. From Community to domestic (I), from domestic to Community (E), from domestic to domestic (D)	I
modByCarrierEnabled	boolean	yes	Indicates whether the carrier can modify the operation or not. When yes, it can be modified, when no, it cannot.	true
carrier/carrierText	30/200 long text	no	Not compulsory. When given, at least one is compulsory	carrierText="Pelda Trans Kft."
sellerName	200 long text	no (in case of tradeType E and D yes)	Name of the sender/seller company, from which product movement is started.	„Első Kereskedő Kft.”
sellerVatNumber	15 long text	no (in case of tradeType E and D yes)	In case of a Hungarian sender, the first 8 digits of the Hungarian tax registration number. In case of a foreign sender, the community tax registration number.	32165478
sellerCountry	2 long text	no (in case of tradeType E and D yes)	Country code of the sender/seller	HU
sellerAddress	200 long text	no (in case of tradeType E and D yes)	Address of the sender/seller	Budapest Kisdobos tér 2.

destinationName	200 long text	no (in case of tradeType I yes)	Name of the destination/buyer company, where the product movement starts.	„Első Kereskedő Kft.”
destinationVatNumber	15 long text	no (in case of tradeType I yes)	Tax registration number of the destination/buyer company. In case of a Hungarian destination, the first 8 digits of the Hungarian tax registration number. In case of a foreign destination, the community tax registration number.	32165478
destinationCountry	2 long text	no (in case of tradeType I yes)	Country code of the destination/buyer company	HU
destinationAddress	200 long text	no (in case of tradeType I yes)	Address of the destination/buyer company	Budapest Kisdobos tér 1.
loadLocation	element	no (in case of tradeType E and D yes)	Address of loading	Budapest Ipartelep u 1.
unloadLocation	element	no (in case of tradeType I and D yes)	Address of unloading	Budapest Közraktár utca 1.
vehicle/plateNumber	element (vehicle data) plate number	no (but has to be filled it before the finalization of the element)	Registration plate number of the vehicle (tractor)	ABC321
vehicle/country	3 long text	no	Country belonging to the registration plate number. Accepted from A to Z.	H
vehicle2/plateNumber	element (vehicle data)	no	The first tow	FFF397
vehicle2/country	3 long text	no	Country belonging to the registration plate number. Accepted from A to Z.	H
vehicle3/plateNumber	element (vehicle data)	no	The second tow, if exists	PDF397
vehicle3/country	3 long text	no	Country belonging to the registration plate number. Accepted from A to Z.	H

loadDate	xsd dateTime	no	Time of load	2014-12-04T08:45:00+01
arrivalDate	xsd dateTime	no (but has to be filled it before the finalization of the element)	Time of unload	2014-12-05T21:15:00+01
items	Element list (tradeCardItem)	yes	Items of the operation. A list containing at least one element.	

2.3.2.2 VERIFICATION OF THE DATA GIVEN IN THE OPERATION

The items list has to contain at least one item.

When tradeType = I (from Community to domestic): Filling in the seller* fields (sender/seller data) is optional, filling in the destination fields (destination/buyer data) is compulsory (destinationCountry=HU), the destination VatNumber must contain a Hungarian tax registration number (8 long), or Hungarian tax ID (10 long).

When tradeType = E (from domestic to Community): Filling in the destination* fields (destination/buyer data) is optional, filling in the seller* fields (sender/seller data) is compulsory (sellerCountry=HU), the sellerVatNumber must contain a Hungarian tax registration number (8 long), or Hungarian tax ID (10 long).

Filling in vehicle element (licence plate of towing/tractor vehicle) is compulsory.

When tradeType = D (domestic -> domestic relation): Filling in the destination* and seller* fields are compulsory and must be of Hungary. The tax number fields must contain Hungarian tax registration number or Hungarian tax ID.

Filling in vehicle element (licence plate of towing/tractor vehicle) is compulsory.

Before finalizing/closing the operation, the following data shall be submitted:

- The vehicle element has to contain actual vehicle data.
- The arrivalDate has to contain the time of the unload.

Further information about verification of the address data is available in [2.3.2.4](#).

2.3.2.3 STRUCTURE AND FIELDS OF THE LOAD AND UNLOAD ADDRESS DATA ELEMENT

Field name	Type	Compulsory	Description	Example
name	150 long text	yes	Name of the company at the address. Manager or owner of warehouse.	Raktarozó kft.
VATNumber	15 long text	no, (when country=HU, is compulsory)	In case of a Hungarian company, the first 8 digits of the Hungarian tax registration number. In case of a foreign company, the community tax registration number.	24653422
phone	15 long text	no	Telephone number of the warehouse/headquarters	+36221321654
email	128 long text	no	Email of the warehouse/headquarters	info@raktar.hu
country	2 long text	yes	Country code	HU
zipCode	7 long text	yes	Zip code	1111
city	50 long text	yes	City	Budapest
street	150 long text	yes	street	Fő
streetType	50 long text	no (can be given in street field)	Street type	utca
streetNumber	50 long text	no	number	1
lotNumber	15 long text	no	topographical lot number. If house number is unknown or not distributed etc.	11231/A.

In case of load and unload address when the house number is not known or non-existing, the topographical lot number has to be given in the lotNumber field.

2.3.2.4 ADDRESS DATA VERIFICATION

- **When tradeType=I (from Community to domestic relation):** Address of load is optional, unload address compulsory and has to be a Hungarian address, tax registration number (8 long) given at unload address has to be an existing Hungarian tax registration number.
- **When tradeType=E (from domestic to Community relation):** Address of load is compulsory, tax registration number (8 long) given at the address has to be an existing Hungarian tax registration number with a Hungarian address. Unload address is optional.
- **When tradeType=D (domestic -> domestic relation):** Both load and unload address are compulsory. Addresses given have to be Hungarian, and the tax registration numbers (8 long) given have to be existing Hungarian tax registration numbers.

In case of transport from Community to domestic destination, when the operation contains a qualified product which can only be imported by companies having a FELÍR (Supervisory Information System) number issued by NÉBIH (National Food Chain Safety Office), the unload address can only be an address registered in the registry of NÉBIH.

2.3.2.5 COUNTRY LIST

At the address data and the country fields only the following country codes can be given:

- AT Austria
- BE Belgium
- BG Bulgaria
- CY Cyprus
- CZ Czech Republic
- DK Denmark
- GB United Kingdom
- EE Estonia
- FI Finland
- FR France
- GR Greece
- NL the Netherlands
- HR Croatia
- IE Ireland
- PL Poland
- LT Lithuania
- LV Latvia
- LU Luxembourg
- HU Hungary
- MT Malta
- DE Germany
- IT Italy
- PT Portugal
- RO Romania
- ES Spain
- SE Sweden
- SK Slovakia
- SI Slovenia

2.3.2.6 ITEMS LIST STRUCTURE (TRADECARDITEM)

The items list contains tradeCardItems, which describe items related to the operation. The item contains the product(s) related to the transport, their weight and other information.

One item is related to the following data:

Field name	Type	Compulsory	Description	Example
id	Attribute 30 long ID	no	Generated by the server at the setting of the operation. In case of Create operation not to be filled in. In case of modification compulsory. The server identifies the item to be modified based on this data.	12ASDF356DFG
tradeReason	Enumerated: S: Sale A: Achieved W: Wagework O: Other	yes	Reason to transport the goods.	S
productVtsz	20 long text	yes	VTSZ number of the item.	03034921
productName	200 long text	yes	Product name in text used by the operator. (does not correspond to the VTSZ)	Kékúszójú tonhal filé
adrNumber	10 long text	Only compulsory in case of dangerous goods (e.g. fuel)	Label, in case of transportation of dangerous goods, the value of categorization.	2.1
transportLicense	30 long text	no	In case of transportation of dangerous goods, number of the licence. Issued by the disaster protection authority.	
weight	xs:decimal	yes	Item weight: gross weight in kgs.	425
value	xs:decimal	yes	Procurement value of item in HUF. When the financial settlement is in foreign currency, calculated on the current fx rate known.	12500000 *
factoryItemNumber	30 long text	no	Factory number, when the item covers only one specific product.	7622210240200

importerItemNumber	30 long text	no	Item number used by the operator. When the item A covers only one specific product.	TS7622/11
expirationDate	xs:date	no	When the item is food, the date of expiry.	2015-07-20
batchNumber	30 long	no	Batch number. Production ID.	234

- Value definition:
In case the road transportation of the product is product procurement or product sale, the value of the individual items without tax, in case of road transportation for other purposes, purchase price belonging to the given items without tax, or the purchase price of a similar product without tax, if no such price is available, production price without tax.

2.3.2.7 VERIFICATIONS RELATED TO ITEMS

The system performs verifications at the registration of items based on the following:

- VTSZ number verification (whether it exists).
- Based on VTSZ, when the product is risky, an assurance is calculated. The item can only be recorded when there is sufficient cover for the assurance.
- Based on VTSZ, the product is obliged to have a FELIR number. The FELIR number verification of the operator is performed (based on NÉBIH data). These products can be unloaded only at an address registered in the address list managed by NÉBIH (First warehousing place).
- Based on VTSZ, whether the product belongs to dangerous goods.
- VTSZ numbers of risky and dangerous products shall be given in 8 long.
- Verification of the first warehousing place
- Whether the given good is food.

On item level in case of Active operations, only the following fields can be modified:

- Volume, weight (Kg)
- Value (HUF)
- Plate number, vehicle data (Plate number, vehicle county)
- Unload place data (Unload place address data)

2.3.2.8 TRADEREASON

At the operation items, the reason for recording the given item is to be given. This also affects assurance calculation. The reason for item transportation has to be given in the tradeReason field.

S: Sale of goods. Assurance is calculated.

A: Purchase of goods. Assurance is calculated.

W: Wagerwork. No assurance is calculated.

O: Other purpose. Assurance is calculated.

2.4 MANAGETRADECARDSRESPONSE, STRUCTURE OF THE RESPONSE

On the manageTradeCardsRequest XML sent in as a request the system provides a response XML, which is described by the manageTradeCardsResponse element in the XSD. The response XML contains the result of the processing.

The response XML has the same header and user header as the request.

The business response is within the tradeCardOperationsResults element, which is an operationResult list. The list has as many elements as the request. For any operation arrived in request this list returns the result.

2.4.1 OPERATIONRESULT RESULT

The list received as response contains operationResult elements. One element contains the result of an operation received in one request.

Fields:

Field name	Type	Compulsory	Description	Example
result	OperationResultType xsd type	yes	Contains the result of the operation.	
tradeCardInfo	TradeCardBasicInfoType	yes	Basic data of the operation related to the action.	

The result contains data and success related to the operation.

The tradeCardInfo contains information related to the registration of the operation.

2.4.1.1 RESULT STRUCTURE (OPERATIONRESULTTYPE)

Result element fields:

Field name	Type	Compulsory	Description	Example
funcCode	Enumerated, OK, WARNING, ERROR	yes	Indicates the success of the operation. OK: Everything successful, WARNING: partly successful (typically not in use) ERROR: Error, operation failed	
reasonCode	Enumerated type	yes	Exact error code of the result of the execution. SUCCESS is successful. Others refer to errors. E.g.:	
msg	200 long text	no	In case of error, the msg contains the more detailed text description of the error.	
index	xs:integer xsd whole number type	yes	Ordinal number of the operation (in the request), the result is contained by the operationResult	
operation	Enumerated:create, modify, delete, finalize	yes	Indicates the manner of modification. The type of the given modification task.	create

Index and operation are copied from the operation received in the request. Based on these it can be seen to which operation the given response belongs to in the request.

The reason of the execution can be seen in the funcCode and the reasonCode, while if there was an error, the msg field contains the text description.

2.4.1.2 TRADECARDINFO ELEMENT STRUCTURE

In the response XML this element contains trade data related to the operation (after the execution of the operation, about the current status). The majority also arrived in the request.

Field name	Type	Compulsory	Description	Example
tcn	20 long text	compulsory in case of modify operation, otherwise can be omitted	EKAER number of the operation. Identifies the operation.	12312312331
orderNumber	50 long text	no	Identifies the operation/order in the own system of the user	ASDF234ffas3
tradeType	Enumerated: E, I, D	yes	This field defines the relation of the product movement.	
From Community to	I			

domestic (I), from domestic to Community (E),from domestic to domestic (D)				
modByCarrierE nabled	boolean	yes	Indicates whether the carrier can modify the operation or not. When yes, it can be modified, when no, it cannot.	true
carrier/carrierT ext	30/200 long text	no	Not compulsory. When given, at least one is compulsory	carrierText="Pelda Trans Kft."
sellerName	200 long text	no (in case of tradeType E and D yes)	Name of the sender/seller company, from which product movement is started.	„Első Kereskedő Kft.”
sellerVatNumb er	15 long text	no (in case of tradeType E and D yes)	In case of a Hungarian sender, the first 8 digits of the Hungarian tax registration number. In case of a foreign sender, the community tax registration number.	32165478
sellerCountry	2 long text	no (in case of tradeType E and D yes)	Country code of the sender/seller	HU
sellerAddress	200 long text	no (in case of tradeType E and D yes)	Address of the sender/seller	Budapest Kisdobos tér 2.
destinationNam e	200 long text	no (in case of tradeType I yes)	Name of the destination/buyer company, where the product movement starts.	„Első Kereskedő Kft.”
destinationVat Number	15 long text	no (in case of tradeType I yes)	Tax registration number of the destination/buyer company. In case of a Hungarian destination, the first 8 digits of the Hungarian tax registration number. In case of a foreign destination, the community tax registration number.	32165478
destinationCou ntry	2 long text	no (in case of tradeType I yes)	Country code of the destination/buyer company	HU
destinationAdd ress	200 long text	no (in case of tradeType I	Address of the destination/buyer company	Budapest Kisdobos tér 1.

		yes)		
loadLocation	element	no (in case of tradeType E and D yes)	Address of loading	Budapest Ipartelep u 1.
unloadLocation	element	no (in case of tradeType I and D yes)	Address of unloading	Budapest Közraktár utca 1.
vehicle/plateNumber	element (vehicle data) plate number	no (but has to be filled it before the finalization of the element)	Registration plate number of the vehicle (tractor)	ABC321
vehicle/country	3 long text	no	Country belonging to the registration plate number. Accepted from A to Z.	H
vehicle2/plateNumber	element (vehicle data)	no	The first tow	FFF397
vehicle2/country	3 long text	no	Country belonging to the registration plate number. Accepted from A to Z.	H
vehicle3/plateNumber	element (vehicle data)	no	The second tow, if exists	PDF397
vehicle3/country	3 long text	no	Country belonging to the registration plate number. Accepted from A to Z.	H
loadDate	xsd dateTime	no	Time of load	2014-12-04T08:45:00+01
items	Element list (tradeCardItem)	yes	Items of the operation. A list containing at least one element.	structure described in chapter 2.3.2.5
VATNumber	8 long text	No, only if the registering party has a tax number	Tax number of the registering party. Server side manages and fills automatically.	32165498
taxIdentifier	10 long text	No, only if the registering party has a tax ID	Tax ID of the registering party. Server side manages and fills automatically.	321654879
status	Enumerated: P: under planning S: Active EKAER, received number F: Finalized,	Yes	Current status of the operation. In XML-based operations S status given automatically. P status is given only when registering	S

	finished I: Inactive D: Deleted		via WEB.	
totalWeight	xs:decimal	Yes	Total weight of items registered in the operation in kilograms.	1500
totalValue	xs:decimal	Yes	Total value of items registered in the operation in HUF.	1250000
totalAssuranceLocked	xs:decimal	Yes	Value of assurance locked for the operation in HUF.	187500
finalizationTime	xs:dateTime	No	Time of registering the finalization. After unload.	2015-01-15T17:35:00+01
insDate	xs:dateTime	Yes	Time of registering the request.	2015-01-14T10:25:15+01

2.4.1.3 OPERATION STATUSES (STATUS)

The operations have a technical lifecycle managed by the status field. It is given from which status to which status one can step to, and what compliance tests are performed by the system in case of status change. In case of non-compliance the status cannot be modified.

Status codes:

- **P:** Under planning. An operation is put into this status only when it is created through the WEB surface. The operation remains in this status until the user requests an EKAER number for the operation, indicating the planning is over.
- **S:** An active operation with an EKAER number. The unload has not been reported yet, or it is within 15 days. The assurance has been already calculated. Operations created through the XML communication interface are automatically created in this status, therefore are given an EKAER number automatically, and the assurance is also calculated.
- **F:** Finalized operation, either with an expired lifecycle of 15 days, or the fact and time of unload has been already reported.
- **I:** Inactive operation. An operation can be put from S (Active) status to inactive status when deleted. In this case the operation becomes inactive, assurance calculation is performed and as a result the locked assurance limit is set free.
- **D:** Deleted operation. An operation can be put from status P (Under planning) into this status. The operation can be put in P status only as when created through the WEB interface.

2.5 PROCESS AND STEPS OF ASSURANCE CALCULATION

The system manages the assurances in a 60 day “sliding” window. The assurances behind the operations are calculated from the issue of the EKAER number to 60 days retroactively based on the value of the risky products on the operation.

Assurance is only calculated in the following transportation relations:

- Transportation from the Community to domestic, international
- Transportation from domestic to domestic, domestic

Based on the reported value of each product considered risky (by law) an assurance is calculated by the system.

The calculation of the assurance is performed at the same time as the allocation of the EKAER number. In practice, this means in case of a new operation created via XML communication it takes place immediately (as the operation has a status S), and in case of an operation via WEB when edited, as a result of the function “request EKAER number”, (when it is put from status P to status S) the assurance is calculated by the system (assurance calculation and reservation available).

In case of the modification of items of operations in status S when the value of the given item is modified, the system recalculates the necessary assurance automatically. If the modification results in an increase of value, the risk assurance is increased accordingly, and if there is no sufficient assurance limit available, the system does not allow the modification. When the modification is decrease, the risk assurance behind the operation is also decreased.

When an item is deleted, the sum of the risk assurance behind the operation is also set free.

When a operation is put into inactive or deleted status, then assurances related to the risky goods behind it are deleted from the assurance calculation.

2.6 QUERYTRADECARDS, QUERY OPERATIONS

Query can be performed on the operations later. The specification of the operation is still in progress. It will be available in the next specification. The XSD contains the query request and the response, the server serves it, but the text specification is under construction.

3 TECHNICAL DESCRIPTION OF THE SERVICE

3.1 GENERAL TECHNICAL DATA

The appropriate XML has to be sent to the service via the http POST method, and as a result the XML is returned in the response body. The caller defines the operation to be performed in the request, while in the response the server returns the result of the operation.

Context root:

/EkaerManagementService

XSD:

hu\gov\nav\schemas\EKAER\1.0\ekaermanagement.xsd

The XML messages described by the xsd have to be sent to the server via POST method.

Entities used for communication are defined as elements in XSD.

The use and interpretation of the individual elements is also documented in XSD.

3.2 OPERATIONS

- **/customer/manageTradeCards**: Management of operations.
- **/customer/queryTradeCards**: Query of operations.

3.3 HTTP HEADERS

The following http headers have to be given in the request:

content-type=text/xml

accept=text/xml

3.4 HTTP STATUS CODES

The following HTTP status codes will function:

- 200 OK:
The service gives the appropriate business response to the given operation.
- 403 Unauthorized
The service returns the SecurityFault defined in the commonservice.xsd.
- 400 Bad request:
Response to an invalid request based on XSD. The service returns the reason of the error in a TechnicalFaultban.
- 500 Internal server error:
The service can return the fault types defined in commonservice.xsd:
 - backendNotAvailable
 - technicalFault
 - businessFault

The exact definition and description of fault types is available in commonservice.xsd.

3.5 RESULT ELEMENT IN THE RESPONSE MESSAGE

The result element is there in every response message. This always reflects the uniformed success of the business response.

- **funcCode:** can be OK, WARNING, ERROR. It simply shows that the business execution failed, resulted in an error, or in case of “warning” succeeded party (where applicable).
- **reasonCode:** The result code of execution. **Values to be used here are defined in xsd, Enumerated type.**
- **msg:** Text description of the result defined by the reasonCode. More accurate description of the error. In case of successful execution not to be filled in, can be omitted.
- **stackTrace:** This element can only be used in a DEV or TEST environment. In this, the server side can transmit the callstack of the error to the client, which makes troubleshooting easier for both parties (server and client).

3.5.1 ReasonCode Enumerated types

The ReasonCode types are continuously extended. At present they still not contain the error codes given to certain more specific validation errors, but they are continuously extended.

The XSD contains a description for the following Enumerated types. The following types and resultCodes has to be interpreted in accordance with the given business process at each operation. Not every reasonCode has a meaning in every operation, this is only a general list:

- **SUCCESS:** successful execution
- **OPERATION_FAILED:** Execution failed. General error, cannot be listed under other error codes.
- **INVALID_INPUT:** The data content of the request received is inappropriate or insufficient. It does not meet business or other data validation rules.
- **INVALID_REQUEST:** The request received cannot be interpreted. E.g.: The structure of the request received in not well-formed.
- **INVALID_USER_OR_PASSWORD:** Login failed. Invalid user name or password.
- **ACCESS_DENIED:** The caller has no right to call the given operation.
- **OBJECT_NOT_FOUND:** Business object cannot be found. E.g.: In case of a query or a transaction. When the client wishes to perform operation related to a non-existing transaction, etc.
- **REQUESTID_NOT_UNIQUE:** The requestId in the context header is not unique. The description of the context header structure can be found in 2.2.
- **SUCCESS_WITH_WARNING:** General error code when the call request is list-based, and not all items could be executed/managed in the list. Interpreted only in operations when the structure of the request and the response make it possible.
- **TC_ITEM_NOT_FOUND:** The tradeCardOperation has to contain a tradeCard element
- **TC_CREATE_ELEMENT_FOUND:** When created, the tcn element has to be omitted within the tradeCard.
- **TC_BOTH_CARRIER_FOUND:** carrier, carrierText: either one or the other has to be given.
- **TCI_ID_FOUND:** The id attribute has to be omitted within the tradeCardItem.
- **TC_SELLER_NAME_EMPTY:** sellerName: in case of tradeType E and D compulsory
- **TC_SELLER_VAT_NUMBER_EMPTY:** sellerVatNumber: in case of tradeType E and D compulsory
- **TC_SELLER_VAT_NUMBER_ERROR:** sellerVatNumber: not appropriate
- **TC_SELLER_COUNTRY_EMPTY:** sellerCountry: in case of tradeType E and D compulsory

- **TC_SELLER_ADDRESS_EMPTY**: sellerAddress: in case of tradeType E and D compulsory
- **TC_DESTINATION_NAME_EMPTY**: destinationName: in case of tradeType I compulsory
- **TC_DESTINATION_VAT_NUMBER_EMPTY**: destinationVatNumber: in case of tradeType I compulsory
- **TC_DESTINATION_VAT_NUMBER_ERROR**: destinationVatNumber: value not appropriate
- **TC_DESTINATION_COUNTRY_EMPTY**: destinationCountry: in case of tradeType I compulsory
- **TC_DESTINATION_ADDRESS_EMPTY**: destinationAddress in case of tradeType I compulsory
- **TC_LOAD_LOCATION_NOT_FOUND**: loadLocation: in case of tradeType E and D compulsory
- **TC_UNLOAD_LOCATION_NOT_FOUND**: unloadLocation: in case of tradeType I and D compulsory
- **TC_VEHICLE_NOT_FOUND**: vehicle: in case of tradeType E and D compulsory
- **TC_LOCATION_NOT_HUNGARY**: It has to be an address in Hungary
- **TC_LOCATION_NOT_COMPLETE**: At the address data the name, vatNumber, country, zipCode, city, street fields are compulsory.
- **TC_DELETE_ONLY_ACTIVE**: Deletion can be executed only when the operation is still “active”.
- **TC_FINALIZE_VEHICLE_DATA_EMPTY**: vehicle/plateNumber: Has to be filled in before the finalization of the operation
- **TC_FINALIZE_ARRIVAL_DATE_EMPTY**: arrivalDate: Has to be filled in before the finalization of the operation
- **TC_MODIFY_BY_CARRIER_DISABLED**: The carrier cannot modify the operation
- **TCI_DANG_PROD_ADRNUMBER_NOT_FOUND**: addrNumber: in case of dangerous item compulsory
- **TC_DESTINATION_MUST_BE_HUNGARY**: destinationCountry: in case of tradeType I can only be 'HU'
- **TC_SELLER_MUST_BE_HUNGARY**: destinationCountry: in case of Type E can only be 'HU'

4 ANNEX

The XSD describing the service and some example XMLs can be found in the annex.

XSD: ekaermanagement.xsd

The example XMLs cover complete http requests and responses. Besides the XML they contain what http header fields the calls and responses contained.

4.1 EXAMPLE XMLS

The example XMLs can be found on the EKÁER FAQ page.

create1:

This example shows a wrong creation trial. At create operation the tcn element of the tradeCard has to be omitted.

create2:

Example for a successful creation of an operation.

modify1:

Successful operation modification request.

4.2 ACCESSIBILITY OF THE TEST SYSTEM

URL: <https://import-test-b.ekaer.nav.gov.hu/TradeCardService/customer/manageTradeCards>

To access the test system appropriate registration is needed, and the user creating the XML has to have a secret signature code, which is necessary to create the requestSignature.

The service has an operation supporting development, which only performs the validation of the XML, but does not generate a real business process. The structure of the request and the response is identical with the one defined at the operation actually managing the requests. Therefore the TradeCardsRequest defined in xsd expects a TradeCardsRequest element and provides a manageTradeCardsResponse element.

Validation operation URL:

<https://import-test-b.ekaer.nav.gov.hu/TradeCardService/customer/validateTradeCardRequest>