Integration manual 2.0

CONTENTS

DOCUMENT INFORMATION

This document describes all application programming interfaces (API) used in communication which involves the ID Issuer system.

The description of APIs stated in this document is based on the Data Dictionary (version 2.0) and List Of Specifications (version 2.0) released by the secondary repository.

This is not a final version of the document, it is a draft and could be updated due to changes in the previous stated documents.

GENERAL TECHNICAL INFORMATION

The API is offered with an http based RestAPI with JSON parameters. Details of the interfaces offered, and supported messages are defined in this document.

HTTP POST method is used for all calls.

TECHNICAL PARAMETERS:

We have some technical limitations concerning messages:

- The maximum message size that we can accept or send is 8MB
- Limit on the HTTP header size is 10240 bytes
- The maximum response time is 60 seconds

AUTHENTICATION

Our Id Issuer system uses OAuth 2.0 to authorize access to the web service methods. OAuth 2.0 is the industry-standard protocol for authorization. OAuth 2.0 focuses on client developer simplicity while providing specific authorization flows for web applications, desktop applications and server to server communication.

Access tokens are issued as credentials used to access protected resources. An access token is a string representing an authorization issued to the client. The string is opaque to the client and passed in the authentication header. Tokens represent specific scopes and durations of access, granted by the resource owner, and enforced by the resource server and authorization server. Tokens have an expiry of 3600 seconds (1 hour).

GAINING OF THE ACCESS TOKEN:

The way of how to gain the access token is to call URL:

- <<Authentication endpoint>>?grant_type = client_credentials

This URL has to be called with the basic authentication where the username and password will be stated:

- Username: name of the system of economic operator
- Password: password for the system chosen by the economic operator

Successful response will consist of:

- Status code: 200
- Json:

```
{
 "access_token": "<generated token>",
 "expires_in": 3600,
 "token_type": "Bearer"
```

```
}
```

If the username and password are not correct the response will consist of:

- Status code: 401
- Json:

```
{
 "error": "invalid_client"
}
```

CALLING APIS SECURED WITH OAUTH 2:

When calling the APIs secured with the OAuth 2, gained access token must be sent in every request header. The header must contain:

- X-OriginalHash - MD5(message body)
- Content-Type – application/json
- Authorization - <<access_token>>

The receiving system checks the value the X-OriginalHash and compare it to the original message.

If something is wrong with the access token Id Issuer will send an error response with these two types of error message:

HTTP status	Error Code	Error Description
401	SECURITY_INVALID_TOKEN	Invalid security token
401	SECURITY_EXPIRED_TOKEN	Expired security token

Calling the APIs must be sequential not parallel. This means, that if you want to call some API with the same message type (for example ISU) several times in a row, first you have to wait for the response to the first request, than you can send next request.

SECURED COMMUNICATION

Communication between the Id Issuer and interacting participants of the tobacco industry is secured by TSL 1.2 encryption AES256 cypher. Cypher suites that are less secure are not supported.

FAIR USE POLICY

API provided for request for unit level UIs or aggregated level UIs is designed to provide UIs in certain amount. There is no limit on minimal amount of requested UIs, but fair use of API is expected and repeated requests with low volumes in short time period for the same product code shall be prevented.

DATA TYPES

In this part of the document are stated code lists, that are used later in description of the APIs.

COUNTRY CODE LIST

Code	Value
AD	Andorra
AE	United Arab Emirates
AF	Afghanistan
AG	Antigua and Barbuda

AI	Anguilla
AL	Albania
AM	Armenia
AO	Angola
AQ	Antarctica
AR	Argentina
AS	American Samoa
AT	Austria
AU	Australia
AW	Aruba
AX	Aland Islands
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BD	Bangladesh
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BH	Bahrain
BI	Burundi
BJ	Benin
BL	Saint Barthélemy
BM	Bermuda
BN	Brunei Darussalam
BO	Bolivia (Plurinational State of)
BQ	Bonaire, Sint Eustatius and Saba
BR	Brazil
BS	Bahamas
BT	Bhutan
BV	Bouvet Island
BW	Botswana
BY	Belarus
BZ	Belize
CA	Canada
CC	Cocos (Keeling) Islands
CD	Congo, Democratic Republic of the
CF	Central African Republic
CG	Congo
CH	Switzerland

CI	Côte d'Ivoire
CK	Cook Islands
CL	Chile
CM	Cameroon
CN	China
CO	Colombia
CR	Costa Rica
CU	Cuba
CV	Cabo Verde
CW	Curaçao
CX	Christmas Island
CY	Cyprus
CZ	Czechia
DE	Germany
DJ	Djibouti
DK	Denmark
DM	Dominica
DO	Dominican Republic
DZ	Algeria
EC	Ecuador
EE	Estonia
EG	Egypt
EH	Western Sahara
ER	Eritrea
ES	Spain
ET	Ethiopia
FI	Finland
FJ	Fiji
FK	Falkland Islands (Malvinas)
FM	Micronesia (Federated States of)
FO	Faroe Islands
FR	France
GA	Gabon
GB	United Kingdom of Great Britain and Northern Ireland
GD	Grenada
GE	Georgia
GF	French Guiana
GG	Guernsey

GH	Ghana
GI	Gibraltar
GL	Greenland
GM	Gambia
GN	Guinea
GP	Guadeloupe
GQ	Equatorial Guinea
GR	Greece
GS	South Georgia and the South Sandwich Islands
GT	Guatemala
GU	Guam
GW	Guinea-Bissau
GY	Guyana
HK	Hong Kong
HM	Heard Island and McDonald Islands
HN	Honduras
HR	Croatia
HT	Haiti
HU	Hungary
ID	Indonesia
IE	Ireland
IL	Israel
IM	Isle of Man
IN	India
IO	British Indian Ocean Territory
IQ	Iraq
IR	Iran (Islamic Republic of)
IS	Iceland
IT	Italy
JE	Jersey
JM	Jamaica
JO	Jordan
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KH	Cambodia
KI	Kiribati
KM	Comoros
KN	Saint Kitts and Nevis

KP	Korea (Democratic People's Republic of)
KR	Korea, Republic of
KW	Kuwait
KY	Cayman Islands
KZ	Kazakhstan
LA	Lao People's Democratic Republic
LB	Lebanon
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
LY	Libya
MA	Morocco
MC	Monaco
MD	Moldova, Republic of
ME	Montenegro
MF	Saint Martin (French part)
MG	Madagascar
MH	Marshall Islands
MK	Macedonia, the former Yugoslav Republic of
ML	Mali
MM	Myanmar
MN	Mongolia
MO	Macao
MP	Northern Mariana Islands
MQ	Martinique
MR	Mauritania
MS	Montserrat
MT	Malta
MU	Mauritius
MV	Maldives
MW	Malawi
MX	Mexico
MY	Malaysia
MZ	Mozambique

NA	Namibia
NC	New Caledonia
NE	Niger
NF	Norfolk Island
NG	Nigeria
NI	Nicaragua
NL	Netherlands
NO	Norway
NP	Nepal
NR	Nauru
NU	Niue
NZ	New Zealand
OM	Oman
PA	Panama
PE	Peru
PF	French Polynesia
PG	Papua New Guinea
PH	Philippines
PK	Pakistan
PL	Poland
PM	Saint Pierre and Miquelon
PN	Pitcairn
PR	Puerto Rico
PS	Palestine, State of
PT	Portugal
PW	Palau
PY	Paraguay
QA	Qatar
RE	Réunion
RO	Romania
RS	Serbia
RU	Russian Federation
RW	Rwanda
SA	Saudi Arabia
SB	Solomon Islands
SC	Seychelles
SD	Sudan
SE	Sweden
SG	Singapore

SH	Saint Helena, Ascension and Tristan da Cunha
SI	Slovenia
SJ	Svalbard and Jan Mayen
SK	Slovakia
SL	Sierra Leone
SM	San Marino
SN	Senegal
SO	Somalia
SR	Suriname
SS	South Sudan
ST	Sao Tome and Principe
SV	El Salvador
SX	Sint Maarten (Dutch part)
SY	Syrian Arab Republic
SZ	Eswatini
TC	Turks and Caicos Islands
TD	Chad
TF	French Southern Territories
TG	Togo
TH	Thailand
TJ	Tajikistan
TK	Tokelau
TL	Timor-Leste
TM	Turkmenistan
TN	Tunisia
TO	Tonga
TR	Turkey
TT	Trinidad and Tobago
TV	Tuvalu
TW	Taiwan, Province of China
TZ	Tanzania, United Republic of
UA	Ukraine
UG	Uganda
UM	United States Minor Outlying Islands
US	United States of America
UY	Uruguay
UZ	Uzbekistan
VA	Holy See
VC	Saint Vincent and the Grenadines

VE	Venezuela (Bolivarian Republic of)
VG	Virgin Islands (British)
VI	Virgin Islands (U.S.)
VN	Viet Nam
VU	Vanuatu
WF	Wallis and Futuna
WS	Samoa
YE	Yemen
YT	Mayotte
ZA	South Africa
ZM	Zambia
ZW	Zimbabwe

FACILITY TYPE

Value	Name
1	Manufacturing site with warehouse
2	Standalone warehouse
3	First retail outlet
4	Other

RECAL SEASON TYPE

Value	Name
1	Reported event did not materialise
2	Message contained erroneous information
3	Other

TABACCO PRODUCT TYPE

Value	Name
1	Cigarette
2	Cigar
3	Cigarillo
4	Roll your own tobacco
5	Pipe tobacco
6	Waterpipe tobacco
7	Oral tobacco
8	Nasal tobacco
9	Chewing tobacco
10	Novel tobacco product
11	Other

PROVIDED INTERFACE

This chapter contains details of all interfaces provided by Id Issuer system.

The process flow of calling the APIs in right order is shown in next two pictures.

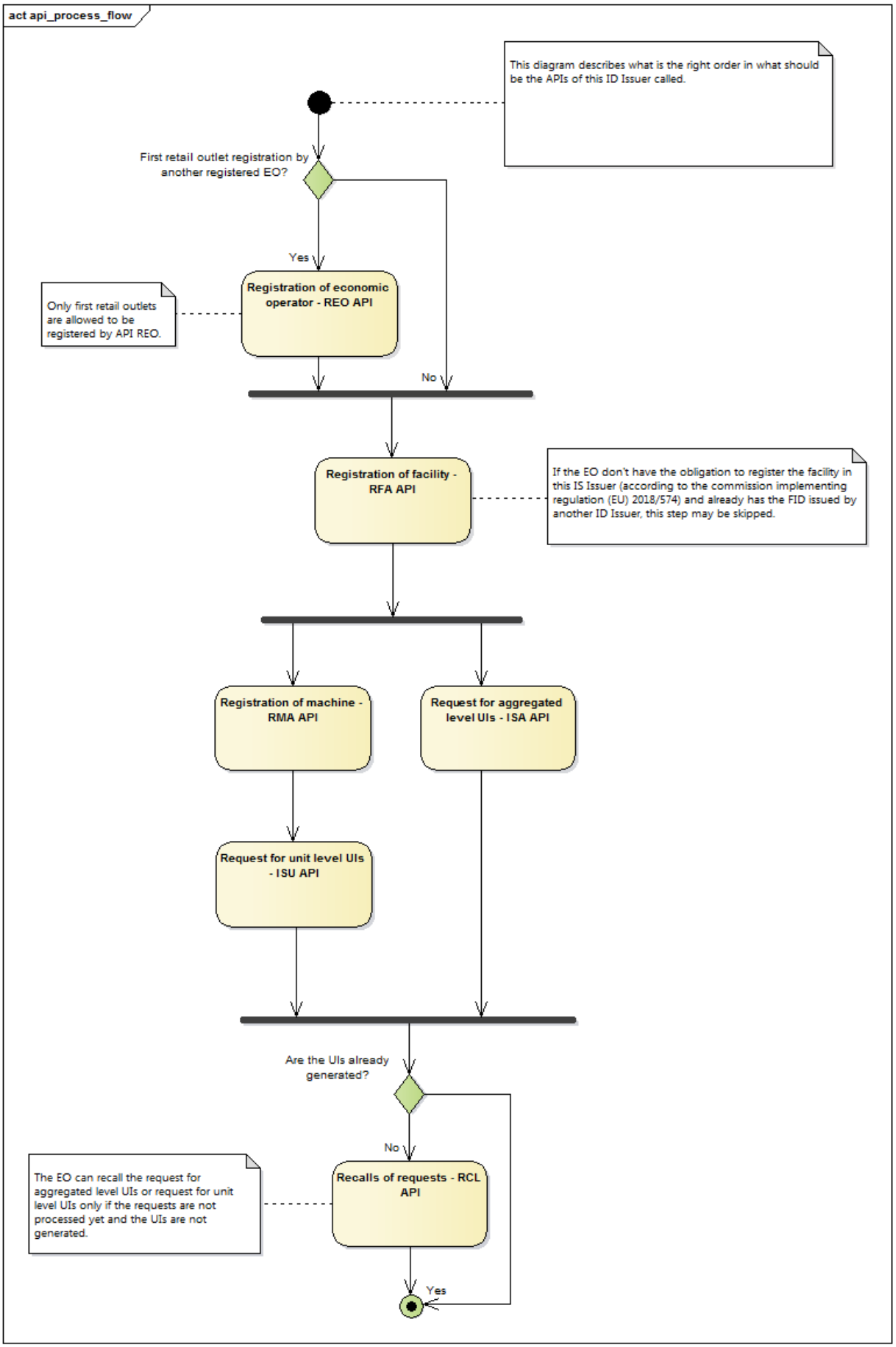

Image 1 - Registration and requests process flow

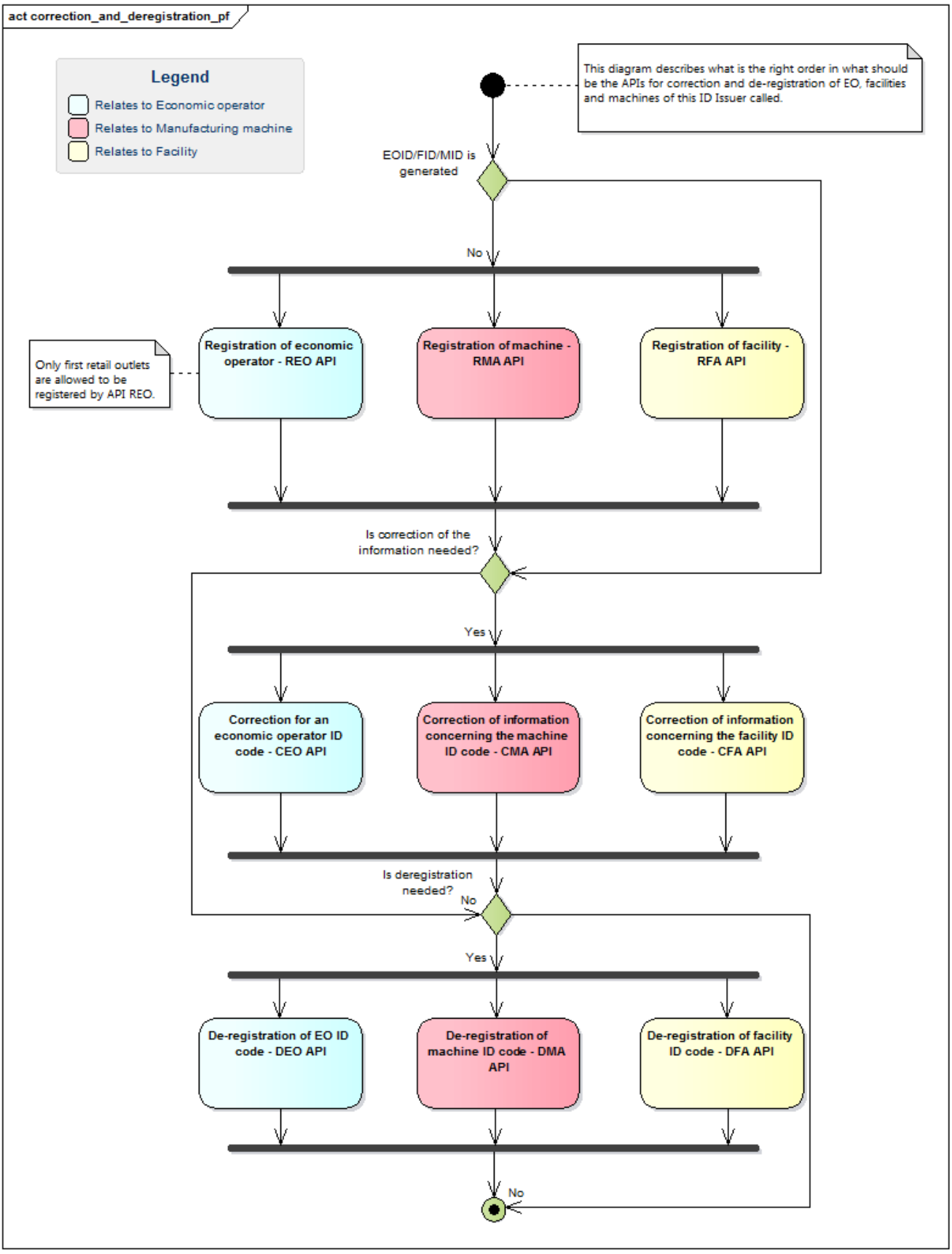

Image 2 - Correction of information and deregistration process flow

REO - REGISTRATION OF AN ECONOMIC OPERATOR

GENERAL DESCRIPTON

Submit the information for the first registration of the economic operator. This interface can be used only to register the first retail outlets by another registered economic operator. This is because an unregistered economic operator cannot have the access token generated.

DESCRIPTION OF THE FIELDS – REQUEST:

Field	Description	Data Type	Cardinality	Priority	Values
EO_Name1	Economic operator's registered name	Text	S (Single value)	M (Mandatory)	
EO_Name2	Economic operator's alternative or abridged name	Text	S (Single value)	O (Optional)	
EO_street	Economic operator's street name and house number (or road number and kilometre)	Text(300)	S (Single value)	M (Mandatory)	
EO_municipality	Economic operator's municipality (city, town or village)	Text(100)	S (Single value)	M (Mandatory)	
EO_postcode	Economic operator's postal code	Text(50)	S (Single value)	M (Mandatory)	'n/a' is permitted value if no postal code has been assigned
EO_A_info	Additional information on economic operator's address (e.g. location in the shopping mall or industrial area)	Text(100)	S (Single value)	O (Optional)	
EO_CountryReg	Economic operator's country of registration	Text(2)	S (Single value)	M (Mandatory)	Code from Country code list

EO_Email	Economic operator's email address; used to inform about registration process, incl. subsequent changes and other required correspondence	Text(80)	S (Single value)	M (Mandatory)	
VAT_R	Indication of the VAT registration status	Boolean	S (Single value)	M (Mandatory)	0 – No VAT registration 1 – VAT number exist
VAT_N	Economic operator's VAT number	Text	S (Single value)	M (Mandatory), if VAT_R = 1	
TAX_N	Economic operator's tax registration number	Text	S (Single value)	M (Mandatory), if VAT_R = 0	
EO_ExciseNumber1	Indication if the economic operator has an excise number issued by the competent authority for the purpose of identification of persons/premises	Boolean	S (Single value)	M (Mandatory)	0 – No SEED number 1 – SEED number exists
EO_ExciseNumber2	Economic operator's excise number issued by the competent authority for the purpose of identification of persons/premises	Text(13)	S (Single value)	M (Mandatory), if EO_Excise Number1 = 1	
OtherEOID_R	Indication if the economic operator has been allocated an identifier by another ID Issuer	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes
OtherEOID_N	Economic operator identifier codes allocated by other ID Issuers	Text(50)	M (Multiple values)	M (Mandatory), if OtherEOI D_R = 1	
Reg_3RD	Indication if the registration is	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes

	made on behalf of a retail outlet operator not otherwise involved in the tobacco trade				
Reg_EOID	Identifier of the economic operator that acts on behalf of a retail outlet operator not otherwise involved in the tobacco trade	Text(50)	S (Single value)	M (Mandatory), if Reg_3RD = 1	
EO_OtherID	Optional identifier	Text(50)	S (Single value)	O (Optional)	
Extensibility	Optional extensibility field	Text(5000)	S (Single value)	O (Optional)	
Message_Type	The identifier of the type of message.	Text	S (Single value)	M (Mandatory)	REO
Code	Unique identifier of the message. Used for recall too	Text	S (Single value)	M (Mandatory)	

Request sample:


```
{
  "EO_Name1": "Example Legal Entity",
  "EO_Name2": "",
  "EO_street": "59 Legal Street",
  "EO_municipality": "Berlin",
  "EO_postcode": "12345",
  "EO_A_info": "Additional info",
  "EO_CountryReg": "DE" ,
  "EO_Email": "email@test.com",
  "VAT_R": 1,
  "VAT_N": "VATNumber 1",
  "TAX_N": "Tax",
  "EO_ExciseNumber1": 1,
  "EO_ExciseNumber2": "LA111FD",
  "OtherEOID_R": 1,
  "OtherEOID_N": [],
  "Reg_3RD": 1,
  "Reg_EOID": "QCALLA01a",
  "EO_OtherID": "1234567890128",
  "Message_Type": "REO",
  "Code": null
}
```


The structure of EO identifier code:

- First 5 characters – ID Issuer prefix (QCALL) according to ISO/IEC 15459-2:2015.5

» Another 4 alphanumeric characters – it is possible that the length of this part of EOID may vary in the future

DESCRIPTION OF THE FIELDS – RESPONSE:

Field	Description	Data Type	Cardinality	Priority	Values
Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	REO
Error	Indicates the failure of the message reception	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes
Errors	Array containing Error_Code, Error_Descr, InternalId	Text	S (Single value)	M (Mandatory) if Error =1	
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	
Checksum	The calculated checksum of the data received	Text	S (Single value)	M (Mandatory)	

MD5 hash function is used: Checksum = MD5(message body)

The response message contains the hash of the original message in the "Checksum" field.

Successful response sample:
HTTP Status 202

```
{
 "Code": " 6854f9a6-a2b2-4c08-8000-0173f3c35567",
 "Message_Type": "REO",
 "Error": 0,
 "Errors": null,
 "Checksum": "G6HF5H"
}
```

Error response:

HTTP status	Error Code	Error Description
401	SECURITY_INVALID_TOKEN	Invalid security token
401	SECURITY_EXPIRED_TOKEN	Expired security token
400	INVALID_REQUEST_FORMAT	This error is returned when at least one of the mandatory fields are missing.
400	INVALID_MESSAGE_TYPE	When the field “Message_Type” is out of the defined list.
400	INVALID_INPUT_FORMAT	When the body of the message doesn’t contain a valid JSON.

400	FAILED_VALIDATION	This error is returned when at least one of the mandatory fields are missing, if an enumeration is wrong
400	ALREADY_EXISTS	Indicated that the CRUD action in add a new entity failed, as the item already exist. This is when checking of the item id already exists.
400	REQUIRED_FIELD_FAILED_VALIDATION	Mandatory fields is missing
400	MAX_LENGTH_ FAILED_VALIDATION	Over max length of field.
400	EOID_NOT_EXIST_OR_ACTIVE	Check if EOID, exists and is active (VAL_ENT_EXIST_EOID, VAL_ENT_ACTIVE_EOID)
400	EXCISE_NUMBER_NOT_VALID	Seed type format wrong.
500	SYSTEM_ERROR	Internal system error.

Error response sample:

The system provides the sufficient details to allow external systems, administrators to identify precisely the issue in order to act accordingly.

The response message can contain a list of error

```
"Errors": [
 { << Error >>},
 { << Error >>},
 { << Error >>},
],
```

Each error contains the following information.

- Error_InternalID is the unique identification of the message processing and validation activity.
- Error_Code is the identifier of the type within the systems.
- Error_Descr is the description in human readable format containing specific error information
- Error_Data is the data for which the error is talking about. This can be used for EO_IDs, F_IDs, M_IDs and UIs

Example of List of errors

```
{
 "Error_InternalID": "yndkFz7TBEO706frD38hzA",
 "Error_Code": "EOID_NOT_EXIST_OR_ACTIVE",
 "Error_Descr": "The EconomicOperatorIdentifier field is unknown.",
 "Error_Data": "QCALL1234"
}
```

Error body sample

```
{
 "Code": " 6854f9a6-a2b2-4c08-8000-0173f3c35567",,
 "Message_Type": null,
 "Error": 1,
 "Errors": [
  {
   "Error_InternalID": "yndkFz7TBEO706frD38hzA",
   "Error_Code":  "INVALID_REQUEST_FORMAT",
   "Error_Descr": "The EconomicOperatorIdentifier field is required."
   "Error_Data": null
```

```
  }
 ],
 "Checksum": "G6HF5H"
}
```


CEO – CORRECTION FOR AN ECONOMIC OPERATOR IDENTIFIER CODE

GENERAL DESCRIPTON

Submit the information of an economic operator known to the repository in order to update 1 or more properties. This information in entirety will over write the previous data held regarding the master data of this economic operator. Links (for example dispatches) to / from this EO_ID will be maintained.

In case that you are registering an economic operator by API, you will always get generated EO_ID and EO_CODE.

DESCRIPTION OF THE FIELDS – REQUEST:

Field	Description	Data Type	Cardinality	Priority	Values
EO_ID	Economic operator identifier code	Text(50)	S (Single value)	M (Mandatory)	
EO_CODE	Economic operator's confirmation code provided in response to the registration of economic operator	Text	S (Single value)	M (Mandatory)	
EO_Name1	Economic operator's registered name	Text	S (Single value)	M (Mandatory)	
EO_Name2	Economic operator's alternative or abridged name	Text	S (Single value)	O (Optional)	
EO_street	Economic operator’s street name and house number (or road number and kilometre)	Text(300)	S (Single value)	M (Mandatory)	
EO_municipality	Economic operator’s municipality (city, town or village)	Text(100)	S (Single value)	M (Mandatory)	

EO_postcode	Economic operator's postal code	Text(50)	S (Single value)	M (Mandatory)	'n/a' is permitted value if no postal code has been assigned
EO_A_info	Additional information on economic operator's address (e.g. location in the shopping mall or industrial area)	Text(100)	S (Single value)	O (Optional)	
EO_CountryReg	Economic operator's country of registration	Text(2)	S (Single value)	M (Mandatory)	Code from Country code list
EO_Email	Economic operator's email address; used to inform about registration process, incl. subsequent changes and other required correspondence	Text(80)	S (Single value)	M (Mandatory)	
VAT_R	Indication of the VAT registration status	Boolean	S (Single value)	M (Mandatory)	0 – No VAT registration 1 – VAT number exist
VAT_N	Economic operator's VAT number	Text	S (Single value)	M (Mandatory), if VAT_R = 1	
TAX_N	Economic operator's tax registration number	Text	S (Single value)	M (Mandatory), if VAT_R = 0	
EO_ExciseNumber1	Indication if the economic operator has an excise number issued by the competent authority for the purpose of identification of persons/premises	Boolean	S (Single value)	M (Mandatory)	0 – No SEED number 1 – SEED number exists
EO_ExciseNumber2	Economic operator's excise	Text(13)	S (Single value)	M (Mandatory),	

	number issued by the competent authority for the purpose of identification of persons/premises			if EO_Excise Number1 = 1	
OtherEOID_R	Indication if the economic operator has been allocated an identifier by another ID Issuer	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes
OtherEOID_N	Economic operator identifier codes allocated by other ID Issuers	Text(50)	M (Multiple values)	M (Mandatory), if OtherEOI D_R = 1	
Reg_3RD	Indication if the registration is made on behalf of a retail outlet operator not otherwise involved in the tobacco trade	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes
Reg_EOID	Identifier of the economic operator that acts on behalf of a retail outlet operator not otherwise involved in the tobacco trade	Text(50)	S (Single value)	M (Mandatory), if Reg_3RD = 1	
EO_OtherID	Optional identifier	Text(50)	S (Single value)	O (Optional)	
Extensibility	Optional extensibility field	Text(5000)	S (Single value)	O (Optional)	
Message_Type	The identifier of the type of message.	Text	S (Single value)	M (Mandatory)	CEO
Code	Unique identifier of the message. Used for recall too	Text	S (Single value)	M (Mandatory)	

Table 1: Description of the fields

Request sample:


```
{
  "EO_ID": "QCALLA01b",
  "EO_CODE": "3404f80caf3d421980d8",
  "EO_Name1": "Example Legal Entity",
  "EO_Name2": "",
  "EO_street": "59 Legal Street",
  "EO_municipality": "Berlin",
```

```
  "EO_postcode": "12345",
  "EO_A_info": "Additional info",
  "EO_CountryReg": "DE" ,
  "EO_Email": "email@test.com",
  "VAT_R": 1,
  "VAT_N": "VATNumber 1",
  "TAX_N": "Tax",
  "EO_ExciseNumber1": 1,
  "EO_ExciseNumber2": "LA111FD",
  "OtherEOID_R": 1,
  "OtherEOID_N": [],
  "Reg_3RD": 1,
  "Reg_EOID": "QCALLA01a",
  "EO_OtherID": "1234567890128",
  "Message_Type": "CEO",
  "Code": null
}
```


DESCRIPTION OF THE FIELDS – RESPONSE:

Field	Description	Data Type	Cardinality	Priority	Values
Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	CEO
Error	Indicates the failure of the message reception	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes
Errors	Array containing Error_Code, Error_Descr, InternalId	Text	S (Single value)	M (Mandatory) if Error =1	
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	
Checksum	The calculated checksum of the data received	Text	S (Single value)	M (Mandatory)	

MD5 hash function is used: Checksum = MD5(message body)

The response message contains the hash of the original message in the "Checksum" field.

Successful response sample:
HTTP Status 202


```
{
 "Code": "873345b2-882f-4064-91f0-90669b46c30a",
 "Message_Type": "CEO",
 "Error": 0,
 "Errors": null,
 "Checksum": "G6HF5H"
}
```

Error response:

HTTP status	Error Code	Error Description
401	SECURITY_INVALID_TOKEN	Invalid security token
401	SECURITY_EXPIRED_TOKEN	Expired security token
400	INVALID_REQUEST_FORMAT	This error is returned when at least one of the mandatory fields are missing.
400	INVALID_MESSAGE_TYPE	When the field “Message_Type” is out of the defined list.
400	INVALID_INPUT_FORMAT	When the body of the message doesn’t contain a valid JSON.
400	FAILED_VALIDATION	This error is returned when at least one of the mandatory fields are missing, if an enumeration is wrong
400	REQUIRED_FIELD_FAILED_VALIDATION	Mandatory fields is missing
400	MAX_LENGTH_ FAILED_VALIDATION	Over max length of field.
400	EOID_NOT_EXIST_OR_ACTIVE	Check if EOID, exists and is active (VAL_ENT_EXIST_EOID, VAL_ENT_ACTIVE_EOID)
400	EXCISE_NUMBER_NOT_VALID	Seed type format wrong.
500	SYSTEM_ERROR	Internal system error.

Error response sample:

The system provides the sufficient details to allow external systems, administrators to identify precisely the issue in order to act accordingly.

The response message can contain a list of error

```
"Errors": [
 { << Error >>},
 { << Error >>},
 { << Error >>},
],
```

Each error contains the following information.

- Error_InternalID is the unique identification of the message processing and validation activity.
- Error_Code is the identifier of the type within the systems.
- Error_Descr is the description in human readable format containing specific error information
- Error_Data is the data for which the error is talking about. This can be used for EO_IDs, F_IDs, M_IDs and UIs

Example of List of errors

```
{
 "Error_InternalID": "yndkFz7TBEO706frD38hzA",
 "Error_Code": "EOID_NOT_EXIST_OR_ACTIVE",
 "Error_Descr": "The EconomicOperatorIdentifier field is unknown.",
 "Error_Data": "QCALL1234"
}
```

Error body sample

```
{
 "Code": " 6854f9a6-a2b2-4c08-8000-0173f3c35567",,
 "Message_Type": null,
 "Error": 1,
 "Errors": [
  {
   "Error_InternalID": "yndkFz7TBEO706frD38hzA",
   "Error_Code":  "INVALID_REQUEST_FORMAT",
   "Error_Descr": "The EconomicOperatorIdentifier field is required."
   "Error_Data": null
  }
 ],
 "Checksum": "G6HF5H"
}
```


DEO - DE-REGISTRATION OF ECONOMIC OPERATOR IDENTIFIER CODE

GENERAL DESCRIPTION

De-registers a previously known operator identifier for a given EO_ID.

DESCRIPTION OF THE FIELDS – REQUEST:

Field	Description	Data Type	Cardinality	Priority	Values
Message_type	The identifier of the type of message.	Text	S (Single value)	M (Mandatory)	DEO
EO_ID	Economic operator identifier code corresponding to the format established by ID issuer coded with the invariant set of ISO646:1991	Text(50)	S (Single value)	M (Mandatory)	
EO_CODE	Economic operator's confirmation code provided in response to the registration of economic operator	Text	S (Single value)	M (Mandatory)	
Reg_3RD	Indication if the registration is made on behalf of a retail outlet operator not otherwise involved in the tobacco trade	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes
Reg_EOID	Identifier of the economic operator that acts on behalf	Text(50)	S (Single value)	M (Mandatory), if	

	of a retail outlet operator not otherwise involved in the tobacco trade			Reg_3RD = 1	
Extensibility	Optional extensibility field	Text(5000)	S (Single value)	O (Optional)	
Code	Unique identifier of the message. Used for recall too	Text	S (Single value)	M (Mandatory)	

Request sample:

```
{
 "Message_Type": "DEO",
 "EO_ID": "QCUKR+1AB020054",
 "EO_CODE": "EO1_CODE",
 "Reg_3RD": 0,
 "Reg_EOID": "Machine Id A",
 "Code": null
}
```

DESCRIPTION OF THE FIELDS – RESPONSE:

Field	Description	Data Type	Cardinality	Priority	Values
Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	CEO
Error	Indicates the failure of the message reception	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes
Errors	Array containing Error_Code, Error_Descr, InternalId	Text	S (Single value)	M (Mandatory) if Error =1	
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	
Checksum	The calculated checksum of the data received	Text	S (Single value)	M (Mandatory)	

MD5 hash function is used: Checksum = MD5(message body)

The response message contains the hash of the original message in the "Checksum" field.

Successful response sample:
HTTP Status 202

```
{
 "Code": "873345b2-882f-4064-91f0-90669b46c30a",
 "Message_Type": "DEO",
 "Error": 0,
 "Errors": null,
 "Checksum": "G6HF5H"
}
```


Error response:

HTTP status	Error Code	Error Description
401	SECURITY_INVALID_TOKEN	Invalid security token
401	SECURITY_EXPIRED_TOKEN	Expired security token
400	INVALID_REQUEST_FORMAT	This error is returned when at least one of the mandatory fields are missing.
400	INVALID_MESSAGE_TYPE	When the field “Message_Type” is out of the defined list.
400	INVALID_INPUT_FORMAT	When the body of the message doesn’t contain a valid JSON.
400	FAILED_VALIDATION	This error is returned when at least one of the mandatory fields are missing, if an enumeration is wrong
400	REQUIRED_FIELD_FAILED_VALIDATION	Mandatory fields is missing
400	MAX_LENGTH_ FAILED_VALIDATION	Over max length of field.
400	EIOD_NOT_EXIST_OR_ACTIVE	Check if EOID, exists and is active (VAL_ENT_EXIST_EOID, VAL_ENT_ACTIVE_EOID)
400	EXCISE_NUMBER_NOT_VALID	SEED type format wrong.
500	SYSTEM_ERROR	Internal system error.

Error response sample:

The system provides the sufficient details to allow external systems, administrators to identify precisely the issue in order to act accordingly.

The response message can contain a list of error

```
"Errors": [
 { << Error >>},
 { << Error >>},
 { << Error >>},
],
```

Each error contains the following information.

- Error_InternalID is the unique identification of the message processing and validation activity.
- Error_Code is the identifier of the type within the systems.
- Error_Descr is the description in human readable format containing specific error information
- Error_Data is the data for which the error is talking about. This can be used for EO_IDs, F_IDs, M_IDs and UIs

Example of List of errors


```
{
 "Error_InternalID": "yndkFz7TBEO706frD38hzA",
```

```
 "Error_Code": "EOID_NOT_EXIST_OR_ACTIVE",
 "Error_Descr": "The EconomicOperatorIdentifier field is unknown.",
 "Error_Data": "QCALL1234"
}
```


Error body sample


```
{
 "Code": " 6854f9a6-a2b2-4c08-8000-0173f3c35567",,
 "Message_Type": null,
 "Error": 1,
 "Errors": [
  {
   "Error_InternalID": "yndkFz7TBEO706frD38hzA",
   "Error_Code":  "INVALID_REQUEST_FORMAT",
   "Error_Descr": "The EconomicOperatorIdentifier field is required."
   "Error_Data": null
  }
 ],
 "Checksum": "G6HF5H"
}
```


RFA - REQUEST FOR A FACILITY IDENTIFIER CODE

GENERAL DESCRIPTION

Add a previously unsent / registered facility. Defined as unseen by the existence of the facility id in the repository.

DESCRIPTION OF THE FIELDS – REQUEST:

Field	Description	Data Type	Cardinality	Priority	Values
EO_ID	Economic operator identifier code corresponding to the format established by ID issuer coded with the invariant set of ISO646:1991	Text(50)	S (Single value)	M (Mandatory)	
EO_CODE	Economic operator's confirmation code provided in response to the registration of economic operator	Text	S (Single value)	M (Mandatory)	
F_street	Facility's street name and house number (or road number and kilometre)	Text(300)	S (Single value)	M (Mandatory)	

F_municipality	Facility's municipality (city, town or village)	Text(100)	S (Single value)	M (Mandatory)	
F_postcode	Facility's postal code	Text(50)	S (Single value)	M (Mandatory)	'n/a' is permitted value if no postal code has been assigned
F_A_info	Additional information on facility's address (e.g. location in the shopping mall or industrial area)	Text(100)	S (Single value)	O (Optional)	
F_Country	Facility's country	Text(2)	S (Single value)	M (Mandatory)	Code from Country code list
F_Type	Type of facility	Integer	S (Single value)	M (Mandatory)	Code from Facility Type
F_Type_Other	Description of other facility type	Text(5000)	S (Single value)	M (Mandatory), if F_Type =4	
F_Status	Indication if a part of the facility has a bonded warehouse status	Boolean	S (Single value)	M (Mandatory)	0 - No 1 - Yes
F_ExciseNumber1	Indication if the facility has an excise number issued by the competent authority for the purpose of identification of persons/premises	Boolean	S (Single value)	M (Mandatory)	0 – No SEED number 1 – SEED number exists
F_ExciseNumber2	Facility's excise number issued by the competent authority for the purpose of identification of persons/premises	Text(13)	S (Single value)	M (Mandatory), if F_Excise Number1 = 1	
OtherFID_R	Indication if the facility has been allocated an identifier by another ID Issuer	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes (possible only for non-EU

					facilities)
OtherFID_N	Facility identifier codes allocated by other ID Issuers	Text(50)	M (Multiple values)	M (Mandatory), if OtherFID_R = 1	
PrevFID_B	Indication if the facility was acquired from another operator and had already a facility identifier code	Boolean	S (Single value)	M (Mandatory)	0 – No (first time registration) 1 – Yes
PrevFID_ID	Previous facility identifier used by the former operator of the facility	Text(50)	S (Single value)	M (Mandatory), if PrevFID_B =1	
Reg_3RD	Indication if the registration is made on behalf of a retail outlet operator not otherwise involved in the tobacco trade	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes
Reg_EOID	Identifier of the economic operator that acts on behalf of a retail outlet operator not otherwise involved in the tobacco trade	Text(50)	S (Single value)	M (Mandatory), if Reg_3RD = 1	
Extensibility	Optional extensibility field	Text(5000)	S (Single value)	O (Optional)	
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	
Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	RFA

Request sample:


```
{
 "EO_ID": "QCUKR+1AB020054",
 "EO_CODE": "EO1_CODE",
 "F_street": "59 Legal Street",
 "F_municipality": "Berlin",
 "F_postcode": "12345",
```

```
 "F_A_info": "Industrial area",
 "F_Country": "DE",
 "F_Type": 2,
 "F_Type_Other": null,
 "F_Status": 0,
 "F_ExciseNumber1": 0,
 "F_ExciseNumber2": null,
 "OtherFID_R": 0,
 "OtherFID_N": [],
 "PrevFID_B": 0,
 "Reg_3RD": 0,
 "Reg_EOID": null,
 "Code": null
 "Message_Type": "RFA"
}
```


DESCRIPTION OF THE FIELDS – RESPONSE:

Field	Description	Data Type	Cardinality	Priority	Values
Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	RFA
Error	Indicates the failure of the message reception	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes
Errors	Array containing Error_Code, Error_Descr, InternalId	Text	S (Single value)	M (Mandatory) if Error =1	
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	
F_ID	Facility's identifier registered	Text(50)	S (Single value)	M (Mandatory)	
Checksum	The calculated checksum of the data received	Text	S (Single value)	M (Mandatory)	

MD5 hash function is used: Checksum = MD5(message body)

The response message contains the hash of the original message in the "Checksum" field.

Successful response sample:

HTTP Status 202


```
{
 "Code": "873345b2-882f-4064-91f0-90669b46c30a",
 "Message_Type": "RFA",
 "Error": 0,
 "Errors": null,
 "F_ID": "QCALLa006",
 "Checksum": "G6HF5H"
```

```
}
```

The structure of facility identifier code:

- First 5 characters – ID Issuer prefix (QCALL) according to ISO/IEC 15459-2:2015.5
- Another 4 alphanumeric characters – it is possible that the length of this part of FID may vary in the future

Error response:

HTTP status	Error Code	Error Description
401	SECURITY_INVALID_TOKEN	Invalid security token
401	SECURITY_EXPIRED_TOKEN	Expired security token
400	FAILED_VALIDATION	Generic validation error. Normally get more detail as below.
400	REQUIRED_FIELD_FAILED_VALIDATION	Mandatory fields is missing
400	MAX_LENGTH_ FAILED_VALIDATION	Over max length of field
400	INVALID_REQUEST_FORMAT	This error is returned when at least one of the mandatory fields are missing.
400	INVALID_MESSAGE_TYPE	When the field “M_Type” is out of the defined list.
400	INVALID_INPUT_FORMAT	When the body of the message doesn’t contain a valid JSON.
500	SYSTEM_ERROR	Internal system error.
400	EIOD_NOT_EXIST_OR_ACTIVE	Check if EOID, exists and is active (VAL_ENT_EXIST_EOID, VAL_ENT_ACTIVE_EOID)
400	FID_NOT_EXIST_OR_ACTIVE	Check if FID, exists and is active (VAL_ENT_EXIST_FID, VAL_ENT_ACTIVE_FID)

Error response sample:

The system provides the sufficient details to allow external systems, administrators to identify precisely the issue in order to act accordingly.

The response message can contain a list of error

```
"Errors": [
 { << Error >>},
 { << Error >>},
 { << Error >>},
],
```

Each error contains the following information.

- Error_InternalID is the unique identification of the message processing and validation activity.
- Error_Code is the identifier of the type within the systems.
- Error_Descr is the description in human readable format containing specific error information
- Error_Data is the data for which the error is talking about. This can be used for EO_IDs, F_IDs, M_IDs and UIs

Example of List of errors

```
{
 "Error_InternalID": "yndkFz7TBEO706frD38hzA",
 "Error_Code": "EOID_NOT_EXIST_OR_ACTIVE",
 "Error_Descr": "The EconomicOperatorIdentifier field is unknown.",
 "Error_Data": "QCALL1234"
}
```


Error body sample


```
{
 "Code": " 6854f9a6-a2b2-4c08-8000-0173f3c35567",,
 "Message_Type": null,
 "Error": 1,
 "Errors": [
  {
   "Error_InternalID": "yndkFz7TBEO706frD38hzA",
   "Error_Code":  "INVALID_REQUEST_FORMAT",
   "Error_Descr": "The EconomicOperatorIdentifier field is required."
   "Error_Data": null
  }
 ],
 "Checksum": "G6HF5H"
}
```


CFA - CORRECTION OF INFORMATION CONCERNING THE FACILITY IDENTIFIER CODE

GENERAL DESCRIPTION

Submit the information of a facility known to the repository in order to update one or more properties. This information in entirety will over write the previous data held regarding the master data of this facility. Links (for example dispatches) to / from this F_ID will be maintained.

DESCRIPTION OF THE FIELDS – REQUEST:

Field	Description	Data Type	Cardinality	Priority	Values
EO_ID	Economic operator identifier code corresponding to the format established by ID issuer coded with the invariant set of ISO646:1991	Text(50)	S (Single value)	M (Mandatory)	
EO_CODE	Economic operator's confirmation code provided in response to the registration of economic operator	Text	S (Single value)	M (Mandatory)	
F_ID	Facility identifier code	Text(50)	S (Single value)	M (Mandatory)	

F_street	Facility's street name and house number (or road number and kilometre)	Text(300)	S (Single value)	M (Mandatory)	
F_municipality	Facility's municipality (city, town or village)	Text(100)	S (Single value)	M (Mandatory)	
F_postcode	Facility's postal code	Text(50)	S (Single value)	M (Mandatory)	'n/a' is permitted value if no postal code has been assigned
F_A_info	Additional information on facility's address (e.g. location in the shopping mall or industrial area)	Text(100)	S (Single value)	O (Optional)	
F_Country	Facility's country	Text(2)	S (Single value)	M (Mandatory)	Code from Country code list
F_Type	Type of facility	Integer	S (Single value)	M (Mandatory)	Code from Facility Type
F_Type_Other	Description of other facility type	Text(5000)	S (Single value)	M (Mandatory), if F_Type =4	
F_Status	Indication if a part of the facility has a bonded warehouse status	Boolean	S (Single value)	M (Mandatory)	0 - No 1 - Yes
F_ExciseNumber1	Indication if the facility has an excise number issued by the competent authority for the purpose of identification of persons/premises	Boolean	S (Single value)	M (Mandatory)	0 – No SEED number 1 – SEED number exists

F_ExciseNumber2	Facility's excise number issued by the competent authority for the purpose of identification of persons/premises	Text(13)	S (Single value)	M (Mandatory), if F_Excise Number1 = 1	
OtherFID_R	Indication if the facility has been allocated an identifier by another ID Issuer	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes (possible only for non-EU facilities)
OtherFID_N	Facility identifier codes allocated by other ID Issuers	Text(50)	M (Multiple values)	M (Mandatory), if OtherFI D_R = 1	
PrevFID_B	Indication if the facility was acquired from another operator and had already a facility identifier code	Boolean	S (Single value)	M (Mandatory)	0 – No (first time registration) 1 – Yes
PrevFID_ID	Previous facility identifier used by the former operator of the facility	Text(50)	S (Single value)	M (Mandatory), if PrevFID_B =1	
Reg_3RD	Indication if the registration is made on behalf of a retail outlet operator not otherwise involved in the tobacco trade	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes
Reg_EOID	Identifier of the economic operator that acts on behalf of a retail outlet operator not otherwise involved in the tobacco trade	Text(50)	S (Single value)	M (Mandatory), if Reg_3RD = 1	
Extensibility	Optional extensibility field	Text(5000)	S (Single value)	O (Optional)	
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	

Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	CFA

Table 1: Description of the fields

Request sample:


```
{
 "EO_ID": "QCUKR+1AB020054",
 "EO_CODE": "EO1_CODE",
 "F_ID": "QCALLa006",
 "F_street": "59 Legal Street",
 "F_municipality": "Berlin",
 "F_postcode": "12345",
 "F_A_info": "Industrial area",
 "F_Country": "DE",
 "F_Type": 2,
 "F_Type_Other": null,
 "F_Status": 0,
 "F_ExciseNumber1": 0,
 "F_ExciseNumber2": null,
 "OtherFID_R": 0,
 "OtherFID_N": [],
 "PrevFID_B": 0,
 "Reg_3RD": 0,
 "Reg_EOID": null,
 "Code": null
 "Message_Type": "CFA"
}
```


DESCRIPTION OF THE FIELDS – RESPONSE:

Field	Description	Data Type	Cardinality	Priority	Values
Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	CFA
Error	Indicates the failure of the message reception	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes
Errors	Array containing Error_Code, Error_Descr, InternalId	Text	S (Single value)	M (Mandatory) if Error =1	
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	
Checksum	The calculated checksum of the data received	Text	S (Single value)	M (Mandatory)	

MD5 hash function is used: Checksum = MD5(message body)

The response message contains the hash of the original message in the "Checksum" field.

Successful response sample:
HTTP Status 202

```
{
  "Code": "873345b2-882f-4064-91f0-90669b46c30a",
  "Message_Type": "CFA",
  "Error": 0,
  "Errors": null,
  "Checksum": "G6HF5H"
}
```

Error response:

HTTP status	Error Code	Error Description
401	SECURITY_INVALID_TOKEN	Invalid security token
401	SECURITY_EXPIRED_TOKEN	Expired security token
400	FAILED_VALIDATION	Generic validation error. Normally get more detail as below.
400	REQUIRED_FIELD_FAILED_VALIDATION	Mandatory fields is missing
400	MAX_LENGTH_ FAILED_VALIDATION	Over max length of field
400	INVALID_REQUEST_FORMAT	This error is returned when at least one of the mandatory fields are missing.
400	INVALID_MESSAGE_TYPE	When the field “M_Type” is out of the defined list.
400	INVALID_INPUT_FORMAT	When the body of the message doesn’t contain a valid JSON.
500	SYSTEM_ERROR	Internal system error.
400	EIOD_NOT_EXIST_OR_ACTIVE	Check if EOID, exists and is active (VAL_ENT_EXIST_EOID, VAL_ENT_ACTIVE_EOID)
400	FID_NOT_EXIST_OR_ACTIVE	Check if FID, exists and is active (VAL_ENT_EXIST_FID, VAL_ENT_ACTIVE_FID)

Error response sample:

The system provides the sufficient details to allow external systems, administrators to identify precisely the issue in order to act accordingly.

The response message can contain a list of error

```
"Errors": [
 { << Error >>},
 { << Error >>},
 { << Error >>},
],
```

Each error contains the following information.

- Error_InternalID is the unique identification of the message processing and validation activity.
- Error_Code is the identifier of the type within the systems.
- Error_Descr is the description in human readable format containing specific error information

- Error_Data is the data for which the error is talking about. This can be used for EO_IDs, F_IDs, M_IDs and UIs

Example of List of errors

```
{
 "Error_InternalID": "yndkFz7TBEO706frD38hzA",
 "Error_Code": "EOID_NOT_EXIST_OR_ACTIVE",
 "Error_Descr": "The EconomicOperatorIdentifier field is unknown.",
 "Error_Data": "QCALL1234"
}
```

Error body sample

```
{
 "Code": " 6854f9a6-a2b2-4c08-8000-0173f3c35567",,
 "Message_Type": null,
 "Error": 1,
 "Errors": [
  {
   "Error_InternalID": "yndkFz7TBEO706frD38hzA",
   "Error_Code":  "INVALID_REQUEST_FORMAT",
   "Error_Descr": "The EconomicOperatorIdentifier field is required."
   "Error_Data": null
  }
 ],
 "Checksum": "G6HF5H"
}
```

DFA - DE-REGISTRATION OF FACILITY IDENTIFIER CODE

GENERAL DESCRIPTION

De-registers a previously known facilitz for a given EO_ID.

DESCRIPTION OF THE FIELDS – REQUEST:

Field	Description	Data Type	Cardinality	Priority	Values
EO_ID	Economic operator identifier code corresponding to the format established by ID issuer coded with the invariant set of ISO646:1991	Text(50)	S (Single value)	M (Mandatory)	
EO_CODE	Economic operator's confirmation code provided in response to the registration of economic operator	Text	S (Single value)	M (Mandatory)	
F_ID	Facility identifier code	Text(50)	S (Single value)	M (Mandatory)	

Reg_3RD	Indication if the registration is made on behalf of a retail outlet operator not otherwise involved in the tobacco trade	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes
Reg_EOID	Identifier of the economic operator that acts on behalf of a retail outlet operator not otherwise involved in the tobacco trade	Text(50)	S (Single value)	M (Mandatory), if Reg_3RD = 1	
Extensibility	Optional extensibility field	Text(5000)	S (Single value)	O (Optional)	
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	Code
Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	DFA

Table 2: Description of the fields

Request sample:

```
{
 "EO_ID": "QCUKR+1AB020054",
 "EO_CODE": "DFA",
 "F_ID": "QCUKR<1AB020054000048",
 "Reg_3RD": 0,
 "Reg_EOID": null,
 "Code": null,
 "Message_Type": "DFA"
}
```

DESCRIPTION OF THE FIELDS – RESPONSE:

Field	Description	Data Type	Cardinality	Priority	Values
Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	DFA
Error	Indicates the failure of the message reception	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes

Errors	Array containing Error_Code, Error_Descr, InternalId	Text	S (Single value)	M (Mandatory) if Error =1	
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	
Checksum	The calculated checksum of the data received	Text	S (Single value)	M (Mandatory)	

MD5 hash function is used: Checksum = MD5(message body)

The response message contains the hash of the original message in the "Checksum" field.

Successful response sample:

HTTP Status 202

```
{
  "Code": "873345b2-882f-4064-91f0-90669b46c30a",
  "Message_Type": "DFA",
  "Error": 0,
  "Errors": null,
  "Checksum": "G6HF5H"
}
```

Error response:

HTTP status	Error Code	Error Description
401	SECURITY_INVALID_TOKEN	Invalid security token
401	SECURITY_EXPIRED_TOKEN	Expired security token
400	FAILED_VALIDATION	Generic validation error. Normally get more detail as below.
400	REQUIRED_FIELD_FAILED_VALIDATION	Mandatory fields is missing
400	MAX_LENGTH_ FAILED_VALIDATION	Over max length of field
400	INVALID_REQUEST_FORMAT	This error is returned when at least one of the mandatory fields are missing.
400	INVALID_MESSAGE_TYPE	When the field “M_Type” is out of the defined list.
400	INVALID_INPUT_FORMAT	When the body of the message doesn’t contain a valid JSON.
500	SYSTEM_ERROR	Internal system error.
400	EIOD_NOT_EXIST_OR_ACTIVE	Check if EOID, exists and is active (VAL_ENT_EXIST_EOID, VAL_ENT_ACTIVE_EOID)
400	FID_NOT_EXIST_OR_ACTIVE	Check if FID, exists and is active (VAL_ENT_EXIST_FID, VAL_ENT_ACTIVE_FID)
400	FID_NOT_RELATED_TO_EOID	Check if EOID FID relation (VAL_ENT_REL_EOID_FID)

Error response sample:

The system provides the sufficient details to allow external systems, administrators to identify precisely the issue in order to act accordingly.

The response message can contain a list of error

```
"Errors": [
 { << Error >>},
 { << Error >>},
 { << Error >>},
],
```

Each error contains the following information.

- Error_InternalID is the unique identification of the message processing and validation activity.
- Error_Code is the identifier of the type within the systems.
- Error_Descr is the description in human readable format containing specific error information
- Error_Data is the data for which the error is talking about. This can be used for EO_IDs, F_IDs, M_IDs and UIs

Example of List of errors

```
{
 "Error_InternalID": "yndkFz7TBEO706frD38hzA",
 "Error_Code": "EOID_NOT_EXIST_OR_ACTIVE",
 "Error_Descr": "The EconomicOperatorIdentifier field is unknown.",
 "Error_Data": "QCALL1234"
}
```

Error body sample

```
{
 "Code": " 6854f9a6-a2b2-4c08-8000-0173f3c35567",,
 "Message_Type": null,
 "Error": 1,
 "Errors": [
  {
   "Error_InternalID": "yndkFz7TBEO706frD38hzA",
   "Error_Code":  "INVALID_REQUEST_FORMAT",
   "Error_Descr": "The EconomicOperatorIdentifier field is required."
   "Error_Data": null
  }
 ],
 "Checksum": "G6HF5H"
}
```

RMA - REQUEST FOR A MACHINE IDENTIFIER CODE

GENERAL DESCRIPTION

Submit the information for the first registration of a machine. Please note that this message is the RMA message that is sent to the id issuer, but with the id issuer M_ID added.

DESCRIPTION OF THE FIELDS – REQUEST:

Field	Description	Data Type	Cardinality	Priority	Values

EO_ID	Economic operator identifier code corresponding to the format established by ID issuer coded with the invariant set of ISO646:1991	Text(50)	S (Single value)	M (Mandatory)	
EO_CODE	Economic operator's confirmation code provided in response to the registration of economic operator	Text	S (Single value)	M (Mandatory)	
F_ID	Facility identifier code	Text(50)	S (Single value)	M (Mandatory)	
PrevMID_B	Indication if the object of this request was already registered, e.g. in relation to another machine identifier code	Boolean	S (Single value)	M (Mandatory)	0 – No (first time registration) 1 – Yes
PrevMID_ID	Previous machine identifier used for the object of this request	Text(50)	S (Single value)	M (Mandatory), if PrevMID_B = 1	
M_entirety	Indication if this request concerns the machine (v. a part of thereof)	Boolean	S (Single value)	M (Mandatory)	0 – No (machine part) 1 – Yes (machine)
P_Producer	Part’s producer	Text(20)	S (Single value)	M (Mandatory), if M_entirety = 0	
P_Model	Part’s model	Text(20)	S (Single value)	M (Mandatory), if M_entirety = 0	

P_Number	Part's serial number	Text(20)	S (Single value)	M (Mandatory), if M_entirety = 0	
P_Mobile	Indication if this part is intended to be used with multiple machines (fixed v. mobile part)	Boolean	S (Single value)	M (Mandatory), if M_entirety = 0	0 – No (fixed part) 1 – Yes (mobile part)
P_ATD1	Indication if an anti-tampering device in the sense of Article 2(7) records the functioning of this part	Boolean	S (Single value)	M (Mandatory), if M_entirety = 0	0 – No 1 – Yes
P_ATD2	Anti-tampering device's serial number	Text(100)	S (Single value)	M (Mandatory), if M_entirety = 0 and P_ATD1 = 1	
P_Description	Part's description explaining its technical function	Text(500)	S (Single value)	O (Optional)	
M_Producer	Machine producer	Text(20)	S (Single value)	M (Mandatory), if M_entirety = 1	
M_Model	Machine model	Text(20)	S (Single value)	M (Mandatory), if M_entirety = 1	
M_Number	Machine serial number	Text(20)	S (Single value)	M (Mandatory), if M_entirety = 1	
M_parts	Indication if the machine consists of multiple separately identifiable parts	Boolean	S (Single value)	M (Mandatory), if M_entirety = 1	
M_plist	List of the identifiable parts	Text(50)	M (Multiple values)	M (Mandatory, limited to 1000 MID), if M_entirety = 1 and M_parts = 1	List of MIDs (parts)

M_ATD	Serial number of the anti-tampering device in the sense of Article 2(7)	Text(100)	S (Single value)	M (Mandatory), if M_entirety = 1 and M_parts = 0	
M_Capacity	Maximum capacity over 24hour production cycle expressed in unit packets	Integer	S (Single value)	M (Mandatory), if M_entirety =1	
(Mandatory)	Optional extensibility field	Text(5000)	S (Single value)	O (Optional)	
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	Code
Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	RMA

Request sample:


```
{
  "EO_ID": "QCUKR+1AB020054",
  "EO_CODE": "EO1_CODE",
  "F_ID": "QCUKR<1AB020054000048",
  "PrevMID_B": 0,
  "M_entirety": 1,
  "M_Producer": "Producer",
  "M_Model": "Model",
  "M_Number": "12346",
  "M_parts": 0,
  "M_ATD": "123456789",
  "M_Capacity": 533,
  "Code": null,
  "Message_Type": "RMA"
}
```


DESCRIPTION OF THE FIELDS – RESPONSE:

Field	Description	Data Type	Cardinality	Priority	Values
Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	RMA
Error	Indicates the failure of the message reception	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes

Errors	Array containing Error_Code, Error_Descr, InternalId	Text	S (Single value)	M (Mandatory) if Error =1	
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	
M_ID	Machine's identifier registered	Text(50)	S (Single value)	M (Mandatory)	
Checksum	The calculated checksum of the data received	Text	S (Single value)	M (Mandatory)	

MD5 hash function is used: Checksum = MD5(message body)

The response message contains the hash of the original message in the "Checksum" field.

Successful response sample:

HTTP Status 202

```
{
  "Code": "873345b2-882f-4064-91f0-90669b46c30a",
  "Message_Type":"RMA",
  "Error": 0,
  "Errors": null,
  "M_ID": "QCALL000H",
  "Checksum": "G6HF5H"
}
```

The structure of manufacturing machine identifier code:

- First 5 characters – ID Issuer prefix (QCALL) according to ISO/IEC 15459-2:2015.5
- Another 4 alphanumeric characters – it is possible that the length of this part of MID may vary in the future

Error response:

HTTP status	Error Code	Error Description
401	SECURITY_INVALID_TOKEN	Invalid security token
401	SECURITY_EXPIRED_TOKEN	Expired security token
400	FAILED_VALIDATION	Generic validation error. Normally get more detail as below.
400	REQUIRED_FIELD_FAILED_VALIDATION	Mandatory fields is missing
400	MAX_LENGTH_ FAILED_VALIDATION	Over max length of field
400	INVALID_REQUEST_FORMAT	This error is returned when at least one of the mandatory fields are missing.
400	INVALID_MESSAGE_TYPE	When the field "M_Type" is out of the defined list.
400	INVALID_INPUT_FORMAT	When the body of the message doesn't contain a valid JSON.
500	SYSTEM_ERROR	Internal system error.
400	EIOD_NOT_EXIST_OR_ACTIVE	Check if EOID, exists and is active (VAL_ENT_EXIST_EOID, VAL_ENT_ACTIVE_EOID)

400	FID_NOT_EXIST_OR_ACTIVE	Check if FID, exists and is active (VAL_ENT_EXIST_FID, VAL_ENT_ACTIVE_FID)
400	FID_NOT_RELATED_TO_EOID	Check if EOID FID relation (VAL_ENT_REL_EOID_FID)

Error response sample:

The system provides the sufficient details to allow external systems, administrators to identify precisely the issue in order to act accordingly.

The response message can contain a list of error

```
"Errors": [
 { << Error >>},
 { << Error >>},
 { << Error >>},
],
```

Each error contains the following information.

- Error_InternalID is the unique identification of the message processing and validation activity.
- Error_Code is the identifier of the type within the systems.
- Error_Descr is the description in human readable format containing specific error information
- Error_Data is the data for which the error is talking about. This can be used for EO_IDs, F_IDs, M_IDs and UIs

Example of List of errors

```
{
 "Error_InternalID": "yndkFz7TBEO706frD38hzA",
 "Error_Code": "EOID_NOT_EXIST_OR_ACTIVE",
 "Error_Descr": "The EconomicOperatorIdentifier field is unknown.",
 "Error_Data": "QCALL1234"
}
```

Error body sample

```
{
 "Code": " 6854f9a6-a2b2-4c08-8000-0173f3c35567",,
 "Message_Type": null,
 "Error": 1,
 "Errors": [
  {
   "Error_InternalID": "yndkFz7TBEO706frD38hzA",
   "Error_Code":  "INVALID_REQUEST_FORMAT",
   "Error_Descr": "The EconomicOperatorIdentifier field is required."
   "Error_Data": null
  }
 ],
 "Checksum": "G6HF5H"
}
```

CMA – CORRECTION OF INFORMATION CONCERNING THE MACHINE IDENTIFIER CODE

GENERAL DESCRIPTION

Submit the information of a machine known to the repository in order to update one or more properties. This information in entirety will over write the previous data held regarding the master data of this machine. Links (for example dispatches) to / from this M_ID will be maintained.

DESCRIPTION OF THE FIELDS – REQUEST:

Field	Description	Data Type	Cardinality	Priority	Values
EO_ID	Economic operator identifier code corresponding to the format established by ID issuer coded with the invariant set of ISO646:1991	Text(50)	S (Single value)	M (Mandatory)	
EO_CODE	Economic operator's confirmation code provided in response to the registration of economic operator	Text	S (Single value)	M (Mandatory)	
F_ID	Facility identifier code	Text(50)	S (Single value)	M (Mandatory)	
M_ID	Machine identifier code	Text(50)	S (Single value)	M (Mandatory)	
PrevMID_B	Indication if the object of this request was already registered, e.g. in relation to another machine identifier code	Boolean	S (Single value)	M (Mandatory)	0 – No (first time registration) 1 – Yes
PrevMID_ID	Previous machine identifier used for the object of this request	Text(50)	S (Single value)	M (Mandatory), if PrevMID_B = 1	
M_entirety	Indication if this request concerns the machine (v. a part of thereof)	Boolean	S (Single value)	M (Mandatory)	0 – No (machine part) 1 – Yes (machine)
P_Producer	Part's producer	Text(20)	S (Single value)	M (Mandatory),	

				if M_entirety = 0	
P_Model	Part's model	Text(20)	S (Single value)	M (Mandatory), if M_entirety = 0	
P_Number	Part's serial number	Text(20)	S (Single value)	M (Mandatory), if M_entirety = 0	
P_Mobile	Indication if this part is intended to be used with multiple machines (fixed v. mobile part)	Boolean	S (Single value)	M (Mandatory), if M_entirety = 0	0 – No (fixed part) 1 – Yes (mobile part)
P_ATD1	Indication if an anti-tampering device in the sense of Article 2(7) records the functioning of this part	Boolean	S (Single value)	M (Mandatory), if M_entirety = 0	0 – No 1 – Yes
P_ATD2	Anti-tampering device's serial number	Text(100)	S (Single value)	M (Mandatory), if M_entirety = 0 and P_ATD1 = 1	
P_Description	Part's description explaining its technical function	Text(500)	S (Single value)	O (Optional)	
M_Producer	Machine producer	Text(20)	S (Single value)	M (Mandatory), if M_entirety = 1	
M_Model	Machine model	Text(20)	S (Single value)	M (Mandatory), if M_entirety = 1	
M_Number	Machine serial number	Text(20)	S (Single value)	M (Mandatory), if M_entirety = 1	
M_parts	Indication if the machine consists of multiple separately identifiable parts	Boolean	S (Single value)	M (Mandatory), if M_entirety = 1	

M_plist	List of the identifiable parts	Text(50)	M (Multiple values)	M (Mandatory, limited to 1000 MID), if M_entirety = 1 and M_parts = 1	List of MIDs (parts)
M_ATD	Serial number of the anti-tampering device in the sense of Article 2(7)	Text(100)	S (Single value)	M (Mandatory), if M_entirety = 1 and M_parts = 0	
M_Capacity	Maximum capacity over 24hour production cycle expressed in unit packets	Integer	S (Single value)	M (Mandatory), if M_entirety =1	
(Mandatory)	Optional extensibility field	Text(5000)	S (Single value)	O (Optional)	
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	Code
Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	CMA

Request sample:

```
{
  "EO_ID": "QCUKR+1AB020054",
  "EO_CODE": "EO1_CODE",
  "F_ID": "QCUKR<1AB020054000048",
  "M_ID": "QCUKR>1AB020054000012",
  "PrevMID_B": 0,
  "M_entirety": 1,
  "M_Producer": "Producer",
  "M_Model": "Model",
  "M_Number": "12346",
  "M_parts": 0,
  "M_ATD": "123456789",
  "M_Capacity": 533,
  "Code": null,
  "Message_Type": "CMA"
}
```

DESCRIPTION OF THE FIELDS – RESPONSE:

Field	Description	Data Type	Cardinality	Priority	Values
Message_type	The identifier of the type of message that the	Text	S (Single value)	M (Mandatory)	CMA

	response refers to.				
Error	Indicates the failure of the message reception	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes
Errors	Array containing Error_Code, Error_Descr, InternalId	Text	S (Single value)	M (Mandatory) if Error =1	
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	
Checksum	The calculated checksum of the data received	Text	S (Single value)	M (Mandatory)	

MD5 hash function is used: Checksum = MD5(message body)

The response message contains the hash of the original message in the "Checksum" field.

Successful response sample:

HTTP Status 202

```
{
  "Code": "873345b2-882f-4064-91f0-90669b46c30a",
  "Message_Type":"CMA",
  "Error": 0,
  "Errors": null,
  "Checksum": "G6HF5H"
}
```

Error response:

HTTP status	Error Code	Error Description
401	SECURITY_INVALID_TOKEN	Invalid security token
401	SECURITY_EXPIRED_TOKEN	Expired security token
400	FAILED_VALIDATION	Generic validation error. Normally get more detail as below.
400	REQUIRED_FIELD_FAILED_VALIDATION	Mandatory fields is missing
400	MAX_LENGTH_ FAILED_VALIDATION	Over max length of field
400	INVALID_REQUEST_FORMAT	This error is returned when at least one of the mandatory fields are missing.
400	INVALID_MESSAGE_TYPE	When the field “M_Type” is out of the defined list.
400	INVALID_INPUT_FORMAT	When the body of the message doesn’t contain a valid JSON.
500	SYSTEM_ERROR	Internal system error.
400	EIOD_NOT_EXIST_OR_ACTIVE	Check if EOID, exists and is active (VAL_ENT_EXIST_EOID, VAL_ENT_ACTIVE_EOID)

400	FID_NOT_EXIST_OR_ACTIVE	Check if FID, exists and is active (VAL_ENT_EXIST_FID, VAL_ENT_ACTIVE_FID)
400	FID_NOT_RELATED_TO_EOID	Check if EOID FID relation (VAL_ENT_REL_EOID_FID)
400	MID_NOT_EXIST_OR_ACTIVE	Check if MID, exists and is active (VAL_ENT_EXIST_MID, VAL_ENT_ACTIVE_MID)
400	MID_NOT_RELATED_TO_FID	Check if FID MID relation (VAL_ENT_REL_FID_MID)

Error response sample:

The system provides the sufficient details to allow external systems, administrators to identify precisely the issue in order to act accordingly.

The response message can contain a list of error

```
"Errors": [
 { << Error >>},
 { << Error >>},
 { << Error >>},
],
```

Each error contains the following information.

- Error_InternalID is the unique identification of the message processing and validation activity.
- Error_Code is the identifier of the type within the systems.
- Error_Descr is the description in human readable format containing specific error information
- Error_Data is the data for which the error is talking about. This can be used for EO_IDs, F_IDs, M_IDs and UIs

Example of List of errors

```
{
 "Error_InternalID": "yndkFz7TBEO706frD38hzA",
 "Error_Code": "EOID_NOT_EXIST_OR_ACTIVE",
 "Error_Descr": "The EconomicOperatorIdentifier field is unknown.",
 "Error_Data": "QCALL1234"
}
```

Error body sample

```
{
 "Code": " 6854f9a6-a2b2-4c08-8000-0173f3c35567",,
 "Message_Type": null,
 "Error": 1,
 "Errors": [
  {
   "Error_InternalID": "yndkFz7TBEO706frD38hzA",
   "Error_Code":  "INVALID_REQUEST_FORMAT",
   "Error_Descr": "The EconomicOperatorIdentifier field is required."
   "Error_Data": null
  }
 ],
 "Checksum": "G6HF5H"
}
```

DMA - DE-REGISTRATION OF MACHINE IDENTIFIER CODE

GENERAL DESCRIPTION

De-registers a previously known machine for a given M_ID.

DESCRIPTION OF THE FIELDS – REQUEST:

Field	Description	Data Type	Cardinality	Priority	Values
EO_ID	Economic operator identifier code corresponding to the format established by ID issuer coded with the invariant set of ISO646:1991	Text(50)	S (Single value)	M (Mandatory)	
EO_CODE	Economic operator's confirmation code provided in response to the registration of economic operator	Text	S (Single value)	M (Mandatory)	
F_ID	Facility identifier code	Text(50)	S (Single value)	M (Mandatory)	
M_ID	Machine identifier code	Text(50)	S (Single value)	M (Mandatory)	
Extensibility	Optional extensibility field	Text(5000)	S (Single value)	O (Optional)	
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	Code
Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	DMA

Request sample:

```
{
 "EO_ID":" QCUKR+1AB020054",
 "EO_CODE": "EO1_CODE",
 "F_ID": "QCUKR<1AB020054000048",
 " M_ID": "QCUKR>1AB020054000012" ,
 "Code": null,
 "Message_Type": "DMA"
}
```

DESCRIPTION OF THE FIELDS – RESPONSE:

Field	Description	Data Type	Cardinality	Priority	Values

Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	DMA
Error	Indicates the failure of the message reception	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes
Errors	Array containing Error_Code, Error_Descr, InternalId	Text	S (Single value)	M (Mandatory) if Error =1	
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	
Checksum	The calculated checksum of the data received	Text	S (Single value)	M (Mandatory)	

MD5 hash function is used: Checksum = MD5(message body)

The response message contains the hash of the original message in the "Checksum" field.

Successful response sample:

HTTP Status 202

```
{
  "Code": "873345b2-882f-4064-91f0-90669b46c30a",
  "Message_Type":"DMA",
  "Error": 0,
  "Errors": null,
  "Checksum": "G6HF5H"
}
```

Error response:

HTTP status	Error Code	Error Description
401	SECURITY_INVALID_TOKEN	Invalid security token
401	SECURITY_EXPIRED_TOKEN	Expired security token
400	FAILED_VALIDATION	Generic validation error. Normally get more detail as below.
400	REQUIRED_FIELD_FAILED_VALIDATION	Mandatory fields is missing
400	MAX_LENGTH_ FAILED_VALIDATION	Over max length of field
400	INVALID_REQUEST_FORMAT	This error is returned when at least one of the mandatory fields are missing.
400	INVALID_MESSAGE_TYPE	When the field “M_Type” is out of the defined list.
400	INVALID_INPUT_FORMAT	When the body of the message doesn’t contain a valid JSON.
500	SYSTEM_ERROR	Internal system error.

400	EIOD_NOT_EXIST_OR_ACTIVE	Check if EOID, exists and is active (VAL_ENT_EXIST_EOID, VAL_ENT_ACTIVE_EOID)
400	FID_NOT_EXIST_OR_ACTIVE	Check if FID, exists and is active (VAL_ENT_EXIST_FID, VAL_ENT_ACTIVE_FID)
400	FID_NOT_RELATED_TO_EOID	Check if EOID FID relation (VAL_ENT_REL_EOID_FID)
400	MID_NOT_EXIST_OR_ACTIVE	Check if MID, exists and is active (VAL_ENT_EXIST_MID, VAL_ENT_ACTIVE_MID)
400	MID_NOT_RELATED_TO_FID	Check if FID MID relation (VAL_ENT_REL_FID_MID)

Error response sample:

The system provides the sufficient details to allow external systems, administrators to identify precisely the issue in order to act accordingly.

The response message can contain a list of error

```
"Errors": [
 { << Error >>},
 { << Error >>},
 { << Error >>},
],
```

Each error contains the following information.

- Error_InternalID is the unique identification of the message processing and validation activity.
- Error_Code is the identifier of the type within the systems.
- Error_Descr is the description in human readable format containing specific error information
- Error_Data is the data for which the error is talking about. This can be used for EO_IDs, F_IDs, M_IDs and UIs

Example of List of errors

```
{
 "Error_InternalID": "yndkFz7TBEO706frD38hzA",
 "Error_Code": "EOID_NOT_EXIST_OR_ACTIVE",
 "Error_Descr": "The EconomicOperatorIdentifier field is unknown.",
 "Error_Data": "QCALL1234"
}
```

Error body sample

```
{
 "Code": " 6854f9a6-a2b2-4c08-8000-0173f3c35567",,
 "Message_Type": null,
 "Error": 1,
 "Errors": [
  {
   "Error_InternalID": "yndkFz7TBEO706frD38hzA",
   "Error_Code":  "INVALID_REQUEST_FORMAT",
   "Error_Descr": "The EconomicOperatorIdentifier field is required."
   "Error_Data": null
  }
 ],
```

```
 "Checksum": "G6HF5H"
}
```


ISU - REQUEST FOR UNIT LEVEL UIS

GENERAL DESCRIPTION

Request for reporting the issuance of serial numbers at unit packet level.

DESCRIPTION OF THE FIELDS – REQUEST:

Field	Description	Data Type	Cardinality	Priority	Values
EO_ID	Economic operator identifier code corresponding to the format established by ID issuer coded with the invariant set of ISO646:1991	Text(50)	S (Single value)	M (Mandatory)	
F_ID	Facility identifier code	Text(50)	S (Single value)	M (Mandatory)	
Process_Type	Indication if the production process involves machinery	Boolean	S (Single value)	M (Mandatory)	0 – No (only for fully hand made products) 1 – Yes
M_ID	Machine identifier code	Text(50)	S (Single value)	M (Mandatory), if Process_Type = 1	
P_Type	Type of tobacco product	Integer	S (Single value)	M (Mandatory)	Value from Tabacco Product Type
P_OtherType	Description of other type of tobacco product	Text	S (Single value)	M (Mandatory), if P_Type = 12 (other tobacco product)	
P_CN	Combined Nomenclature (CN) code	Text	S (Single value)	O (Optional)	
P_Brand	Brand of tobacco product under which the product will be marketed on its intended market	Text	S (Single value)	M (Mandatory)	
P_Weight	Average gross weight of unit packet,	Decimal	S (Single value)	M (Mandatory)	

	including packaging, in grams with 0,1 gram accuracy				
P_SubType_Exist	Indicates if the product “subtype name” exists. Subtype name provides further product identification beyond a product's brand name.	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes
P_SubType_Name	The product “subtype name” (if any) as marketed on its intended market	Text	S (Single value)	M (Mandatory), if P_SubType_Exist = 1	
P_units	The number of individual units in the unit packet (number of sticks in the package).	Integer	S (Single value)	M (Mandatory), if P_Type = 1 or 2 or 3	
TP_ID	The identification number of the product used in the EU-CEG system.	Text(14)	S (Single value)	M (Mandatory), if Intended_Market is an EU country	
TP_PN	Tobacco product number used in the EU-CEG system (EAN or GTIN or SKU or UPC)	Text(30)	S (Single value)	M (Mandatory), if Intended_Market is a Member State O (Optional), if Intended_Market is a third country	
Intended_Market	Intended country of retail sale.	Text(2)	S (Single value)	M (Mandatory)	
Intended_Route1	Indication if the product is intended to be moved across country boarders with terrestrial transport.	Boolean	S (Single value)	M (Mandatory)	0 – No 1 - Yes
Intended_Route2	The first country of terrestrial	Text(2)	S (Single value)	M (Mandatory), if Intended_Route1 = 1	

	transport after the product leaves the Member State of manufacturing or the Member State of importation.				
Import	Indication if the product is imported into the EU	Boolean	S (Single value)	M (Mandatory)	0 – No 1 - Yes
Req_Quantity	Requested quantity of aggregated level UIs	Integer	S (Single value)	M (Mandatory)	
P_OtherID	Optional Product ID	Text(20)	S (Single value)	O (Optional)	
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	Code
Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	ISU

Request sample:


```
{
 "EO_ID":"QCUKR+1AB020054",
 "F_ID": "QCUKR<1AB020054000049",
 "Process_Type":1,
 "M_ID":"Machine Id A",
 "P_Type":2,
 "P_OtherType":null,
 "P_Brand":"Product brand A",
 "P_Weight":10.0,
 "P_SubType_Exist":0,
 "P_units":10,
 "TP_ID":" 1234",
 "TP_PN":"1234",
 "Intended_Market":"BG",
 "Intended_Route1":1,
 "Intended_Route2":"BG",
 "Import":0,
 "Req_Quantity":2,
 "P_OtherID":"00012345600012",
 "Code":null,
 "Message_Type": "ISU"
}
```


DESCRIPTION OF THE FIELDS – RESPONSE:

Field	Description	Data Type	Cardinality	Priority	Values
Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	ISU
Error	Indicates the failure of the message reception	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes
Errors	Array containing Error_Code, Error_Descr, InternalId	Text	S (Single value)	M (Mandatory) if Error =1	
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	
Checksum	The calculated checksum of the data received	Text	S (Single value)	M (Mandatory)	

MD5 hash function is used: Checksum = MD5(message body)

The response message contains the hash of the original message in the "Checksum" field.

Successful response sample:
HTTP Status 202

```
{
 "Code": "873345b2-882f-4064-91f0-90669b46c30a",
 "Message_Type": "ISU",
 "Error": 0,
 "Errors": null,
 "Checksum": "G6HF5H"
}
```

Error response:

HTTP status	Error Code	Error Description
401	SECURITY_INVALID_TOKEN	Invalid security token
401	SECURITY_EXPIRED_TOKEN	Expired security token
400	FAILED_VALIDATION	This error is returned when at least one of the mandatory fields are missing, if an enumeration is wrong
400	REQUIRED_FIELD_FAILED_VALIDATION	Mandatory fields is missing
400	MAX_LENGTH_ FAILED_VALIDATION	Over max length of field.
400	INVALID_REQUEST_FORMAT	This error is returned when at least one of the mandatory fields are missing.
400	INVALID_MESSAGE_TYPE	When the field “Message_Type” is out of the defined list.

400	INVALID_INPUT_FORMAT	When the body of the message doesn't contain a valid JSON.
400	EOID_NOT_EXIST_OR_ACTIVE	Check if EOID, exists and is active (VAL_ENT_EXIST_EOID, VAL_ENT_ACTIVE_EOID)
400	FID_NOT_EXIST_OR_ACTIVE	Check if FID, exists and is active (VAL_ENT_EXIST_FID, VAL_ENT_ACTIVE_FID)
400	FID_NOT_RELATED_TO_EOID	Check if EOID FID relation (VAL_ENT_REL_EOID_FID)
400	MID_NOT_EXIST_OR_ACTIVE	Check if MID, exists and is active (VAL_ENT_EXIST_MID, VAL_ENT_ACTIVE_MID)
400	MID_NOT_RELATED_TO_FID	Check if FID MID relation (VAL_ENT_REL_FID_MID)
400	CLAIM_VALIDATION_FAILED	Caller is not allowed to call this method.
500	SYSTEM_ERROR	Internal system error.

Error response sample:

The system provides the sufficient details to allow external systems, administrators to identify precisely the issue in order to act accordingly.

The response message can contain a list of error

```
"Errors": [
 { << Error >>},
 { << Error >>},
 { << Error >>},
],
```

Each error contains the following information.

- Error_InternalID is the unique identification of the message processing and validation activity.
- Error_Code is the identifier of the type within the systems.
- Error_Descr is the description in human readable format containing specific error information
- Error_Data is the data for which the error is talking about. This can be used for EO_IDs, F_IDs, M_IDs and UIs

Example of List of errors

```
{
 "Error_InternalID": "yndkFz7TBEO706frD38hzA",
 "Error_Code": "EOID_NOT_EXIST_OR_ACTIVE",
 "Error_Descr": "The EconomicOperatorIdentifier field is unknown.",
 "Error_Data": "QCALL1234"
}
```

Error body sample

```
{
 "Code": " 6854f9a6-a2b2-4c08-8000-0173f3c35567",,
 "Message_Type": null,
 "Error": 1,
 "Errors": [
  {
   "Error_InternalID": "yndkFz7TBEO706frD38hzA",
```

```
    "Error_Code":  "INVALID_REQUEST_FORMAT",
    "Error_Descr": "The EconomicOperatorIdentifier field is required."
    "Error_Data": null
   }
 ],
 "Checksum": "G6HF5H"
}
```


ISA - REQUEST FOR AGGREGATED LEVEL UIS

GENERAL DESCRIPTION

Request for reporting the issuance of serial numbers at aggregated level.

DESCRIPTION OF THE FIELDS – REQUEST:

Field	Description	Data Type	Cardinality	Priority	Values
EO_ID	Economic operator identifier code corresponding to the format established by ID issuer coded with the invariant set of ISO646:1991	Text(50)	S (Single value)	M (Mandatory)	
F_ID	Facility identifier code	Text(50)	S (Single value)	M (Mandatory)	
Req_Quantity	Requested quantity of aggregated level UIs	Integer	S (Single value)	M (Mandatory)	
Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	ISA
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	Code

Request sample:


```
{
 "EO_ID": " QCUKR+1AB020054",
 "F_ID": "QCUKR<1AB020054000049",
 "Req_Quantity": 2,
 "Message_Type": "ISA",
 "Code":null
}
```


DESCRIPTION OF THE FIELDS – RESPONSE:

Field	Description	Data Type	Cardinality	Priority	Values

Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	ISA
Error	Indicates the failure of the message reception	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes
Errors	Array containing Error_Code, Error_Descr, InternalId	Text	S (Single value)	M (Mandatory) if Error =1	
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	
Checksum	The calculated checksum of the data received	Text	S (Single value)	M (Mandatory)	

MD5 hash function is used: Checksum = MD5(message body)

The response message contains the hash of the original message in the "Checksum" field.

Successful response sample:
HTTP Status 202

```
{
 "Code": "873345b2-882f-4064-91f0-90669b46c30a",
 "Message_Type": "ISA",
 "Error": 0,
 "Errors": null,
 "Checksum": "G6HF5H"
}
```

Error response:

HTTP status	Error Code	Error Description
401	SECURITY_INVALID_TOKEN	Invalid security token
401	SECURITY_EXPIRED_TOKEN	Expired security token
400	FAILED_VALIDATION	This error is returned when at least one of the mandatory fields are missing, if an enumeration is wrong
400	REQUIRED_FIELD_FAILED_VALIDATION	Mandatory fields is missing
400	MAX_LENGTH_ FAILED_VALIDATION	Over max length of field.
400	INVALID_REQUEST_FORMAT	This error is returned when at least one of the mandatory fields are missing.
400	INVALID_MESSAGE_TYPE	When the field “Message_Type” is out of the defined list.
400	INVALID_INPUT_FORMAT	When the body of the message doesn’t contain a valid JSON.

400	EOID_NOT_EXIST_OR_ACTIVE	Check if EOID, exists and is active (VAL_ENT_EXIST_EOID, VAL_ENT_ACTIVE_EOID)
400	FID_NOT_EXIST_OR_ACTIVE	Check if FID, exists and is active (VAL_ENT_EXIST_FID, VAL_ENT_ACTIVE_FID)
400	FID_NOT_RELATED_TO_EOID	Check if EOID FID relation (VAL_ENT_REL_EOID_FID)
400	CLAIM_VALIDATION_FAILED	Caller is not allowed to call this method.
500	SYSTEM_ERROR	Internal system error.

Error response sample:

The system provides the sufficient details to allow external systems, administrators to identify precisely the issue in order to act accordingly.

The response message can contain a list of error

```
"Errors": [
 { << Error >>},
 { << Error >>},
 { << Error >>},
],
```

Each error contains the following information.

- Error_InternalID is the unique identification of the message processing and validation activity.
- Error_Code is the identifier of the type within the systems.
- Error_Descr is the description in human readable format containing specific error information
- Error_Data is the data for which the error is talking about. This can be used for EO_IDs, F_IDs, M_IDs and UIs

Example of List of errors

```
{
 "Error_InternalID": "yndkFz7TBEO706frD38hzA",
 "Error_Code": "EOID_NOT_EXIST_OR_ACTIVE",
 "Error_Descr": "The EconomicOperatorIdentifier field is unknown.",
 "Error_Data": "QCALL1234"
}
```

Error body sample

```
{
 "Code": " 6854f9a6-a2b2-4c08-8000-0173f3c35567",,
 "Message_Type": null,
 "Error": 1,
 "Errors": [
  {
   "Error_InternalID": "yndkFz7TBEO706frD38hzA",
   "Error_Code":  "INVALID_REQUEST_FORMAT",
   "Error_Descr": "The EconomicOperatorIdentifier field is required."
   "Error_Data": null
  }
 ],
 "Checksum": "G6HF5H"
}
```

RCL - RECALLS OF REQUESTS, OPERATIONAL AND TRANSACTIONAL MESSAGES

GENERAL DESCRIPTION

Given a recall id (“Code” in the return of any message) The caller can mark that event invalid. This is possible for message types ISU, ISA).

DESCRIPTION OF THE FIELDS – REQUEST:

Field	Description	Data Type	Cardinality	Priority	Values
EO_ID	Economic operator identifier code corresponding to the format established by ID issuer coded with the invariant set of ISO646:1991	Text(50)	S (Single value)	M (Mandatory)	
Recall_CODE	Message recall code provided to the message sender in the acknowledgement of the original message to be recalled	Text	S (Single value)	M (Mandatory)	
Recall_Reason1	Reason for recalling the original message	Integer	S (Single value)	M (Mandatory)	Value from Recal Season Type
Recall_Reason2	Description of the reason for recalling the original message	Text	S (Single value)	M (Mandatory), if Recall_Reason1 = 3 (other reason)	
Recall_Reason3	Any additional explanations on the reason for recalling the original message	Text	S (Single value)	M (Mandatory)	
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	Code
Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	RCL

Request sample:

```
{
 "EO_ID":" QCUKR+1AB020054",
 "RecallCode":"873345b2-882f-4064-91f0-90669b46c30a",
 "RecallReason1":1,
 "RecallReason2":"Description",
 "RecallReason3":"Comments",
 "Code":null,
 "Message_Type": "RCL"
}
```

DESCRIPTION OF THE FIELDS – RESPONSE:

Field	Description	Data Type	Cardinality	Priority	Values
Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	RCL
Error	Indicates the failure of the message reception	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes
Errors	Array containing Error_Code, Error_Descr, InternalId	Text	S (Single value)	M (Mandatory) if Error =1	
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	
Checksum	The calculated checksum of the data received	Text	S (Single value)	M (Mandatory)	

MD5 hash function is used: Checksum = MD5(message body)

The response message contains the hash of the original message in the "Checksum" field.

Successful response sample:
HTTP Status 202

```
{
 "Code": "873345b2-882f-4064-91f0-90669b46c30a",
 "Message_Type": "RCL",
 "Error": 0,
 "Errors": null,
 "Checksum": "G6HF5H"
}
```

Error response:

HTTP status	Error Code	Error Description
401	SECURITY_INVALID_TOKEN	Invalid security token
401	SECURITY_EXPIRED_TOKEN	Expired security token

400	FAILED_VALIDATION	This error is returned when at least one of the mandatory fields are missing, if an enumeration is wrong
400	REQUIRED_FIELD_FAILED_VALIDATION	Mandatory fields is missing
400	MAX_LENGTH_ FAILED_VALIDATION	Over max length of field.
400	INVALID_REQUEST_FORMAT	This error is returned when at least one of the mandatory fields are missing.
400	INVALID_MESSAGE_TYPE	When the field “Message_Type” is out of the defined list.
400	INVALID_INPUT_FORMAT	When the body of the message doesn’t contain a valid JSON.
400	RECALL_TOO_LATE	If the recall is performed after the 1 business day allowed since the original call.
400	RECALL_AFTER_ONE_WORKING_ DAY	For requests of unit level or aggregated level UIs (ISU, IRU, ISA, IRA), recalls can be performed up to one working day after the original message. (VAL_EVT_RECALL)
500	SYSTEM_ERROR	Internal system error.

Error response sample:

The system provides the sufficient details to allow external systems, administrators to identify precisely the issue in order to act accordingly.

The response message can contain a list of error

```
"Errors": [
 { << Error >>},
 { << Error >>},
 { << Error >>},
],
```

Each error contains the following information.

- Error_InternalID is the unique identification of the message processing and validation activity.
- Error_Code is the identifier of the type within the systems.
- Error_Descr is the description in human readable format containing specific error information
- Error_Data is the data for which the error is talking about. This can be used for EO_IDs, F_IDs, M_IDs and UIs

Example of List of errors

```
{
 "Error_InternalID": "yndkFz7TBEO706frD38hzA",
 "Error_Code": "EOID_NOT_EXIST_OR_ACTIVE",
 "Error_Descr": "The EconomicOperatorIdentifier field is unknown.",
 "Error_Data": "QCALL1234"
}
```

Error body sample

```
{
 "Code": " 6854f9a6-a2b2-4c08-8000-0173f3c35567",,
 "Message_Type": null,
```

```
 "Error": 1,
 "Errors": [
  {
   "Error_InternalID": "yndkFz7TBEO706frD38hzA",
   "Error_Code":  "INVALID_REQUEST_FORMAT",
   "Error_Descr": "The EconomicOperatorIdentifier field is required."
   "Error_Data": null
  }
 ],
 "Checksum": "G6HF5H"
}
```


LBU – GET LIST OF BATCHES FOR UNIT LEVEL UIS REQUEST

GENERAL DESCRIPTION

This API is used for gaining the list of batches that belongs to one concrete request for unit level UIs. It will provide the information if the batches (and UIs) are already generated or not and if they are, the batch names. These batch names will be used in the API for getting the specific UIs from the batch.

This API should not be called with more frequency than once per half an hour (in case the returned information is that the batches are not generated yet).

To get the list of batches, EOID and response code returned in response of calling the ISU API must be sent to the request (element “Req_ul_CODE”).

DESCRIPTION OF THE FIELDS – REQUEST:

Field	Description	Data Type	Cardinality	Priority	Values
EO_ID	Economic operator identifier code of the submitting entity	Text(50)	S (Single value)	M (Mandatory)	
Req_ul_CODE	Message code provided to the message sender in the acknowledgement of the original request for upUIs message. Used for identification of request for upUIs.	Text	S (Single value)	M (Mandatory)	
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	null
Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	LBU

Note that:

» Element “Code” cannot be filled in the request message – please send null value

Request sample:

```
{
 "EO_ID": "QCALLA001",
 "Req_ul_CODE":"581c9a77-b443-4a2f-92e6-f82bbaa2891b",
 "Code": null,
 "Message_Type": "LBU"
}
```

DESCRIPTION OF THE FIELDS – RESPONSE:

Field	Description	Data Type	Cardinality	Priority	Values
Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	LBU
Error	Indicates the failure of the message reception	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes
Errors	Array containing Error_Code, Error_Descr, InternalId	Text	S (Single value)	M (Mandatory) if Error =1	
Gen_complete	Indicates that the generation of batches is complete	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes
Batches	List of batches generated for unit level UIs request	Text	M (Multiple value)	M (Mandatory) if Gen_complete =1	
Req_no	Identifies the request for unit level UIs to which are the batches associated	Text	S (Single value)	M (Mandatory)	
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	
Checksum	The calculated checksum of the data received	Text	S (Single value)	M (Mandatory)	

The response message contains the hash of the original message in the "Checksum" field.

Checksum = X-OriginalHash sent in the original message - request header.

Successful response sample:
HTTP Status 202

```
{
 "Message_Type": "LBU",
 "Code": "873345b2-882f-4064-91f0-90669b46c30a",
 "Error": 0,
 "Errors": null,
 "Gen_complete": 1,
 "Batches": ["BQCALL0000012r","BQCALL000001Io","BQCALL000000WX"],
 "Req_no": "ORD1910000451",
 "Checksum": "G6HF5H"
}
```

Error response:

Set of error responses, that can be sent in this API:

HTTP status	Error Code	Error Description
401	INVALID_OR_EXPIRED_TOKEN	Invalid or expired security token
401	UNAUTHORIZED	Unauthorized call.
400	FAILED_VALIDATION	This error is returned when at least one of the mandatory fields are missing, if an enumeration is wrong
400	INVALID_SIGNATURE	Hash information not matching the message signature.
400	CODE_PROVIDED	Code was provided in context where it's not expected, for example when a manufacturer calls a primary
400	INVALID_MESSAGE_TYPE	When the field "Message_Type" is out of the defined list.
400	INVALID_INPUT_FORMAT	When the body of the message doesn't contain a valid JSON.
400	EIOD_NOT_EXIST_OR_ACTIVE	Check if EOID exists and is active (VAL_ENT_EXIST_EOID, VAL_ENT_ACTIVE_EOID).
400	INVALID_REQUEST_CODE	This error is returned when the sent request code was not found or is not related with sent EO_ID.
500	SYSTEM_ERROR	Internal system error.

Error response sample:

The system provides the sufficient details to allow external systems, administrators to identify the issue precisely in order to act accordingly.

The response message can contain a list of errors

```
"Errors": [
 { << Error >>},
 { << Error >>},
 { << Error >>},
],
```

Each error contains the following information.

- Error_InternalID is the unique identification of the message processing and validation activity.
- Error_Code is the identifier of the type within the systems.
- Error_Descr is the description in human readable format containing specific error information
- Error_Data is closer specification of why the error has occurred

Example of List of errors

```
{
 "Error_InternalID": "257943d2-9437-4eab-bad4-424a24e4ce81",
 "Error_Code": "EOID_NOT_EXIST_OR_ACTIVE",
 "Error_Descr": "EO_ID not exist or active.",
 "Error_Data": "EO_ID or EO_CODE not exist or active."
}
```

Error body sample

```
{
 "Code": "6854f9a6-a2b2-4c08-8000-0173f3c35567",,
 "Message_Type": null,
 "Error": 1,
 "Errors": [
  {
   "Error_InternalID": "257943d2-9437-4eab-bad4-424a24e4ce81",
   "Error_Code":  " FAILED_VALIDATION",
   "Error_Descr": " Not allowed format"
   "Error_Data": "Field 'EO_ID' is mandatory."
  }
 ],
 "Checksum": "G6HF5H"
}
```

LBA – GET LIST OF BATCHES FOR AGGREGATED LEVEL UIS REQUEST

GENERAL DESCRIPTION

This API is used for gaining the list of batches that belongs to one concrete request for aggregated level UIs. It will provide the information if the batches (and UIs) are already generated or not and if they are, the batch names. These batch names will be used in the API for getting the specific UIs from the batch.

This API should not be called with more frequency than once per half an hour (in case the returned information is that the batches are not generated yet).

To get the list of batches, EOID and response code returned in response of calling the ISA API must be sent to the request (element “Req_al_CODE”).

DESCRIPTION OF THE FIELDS – REQUEST:

Field	Description	Data Type	Cardinality	Priority	Values
EO_ID	Economic operator identifier code of the submitting entity	Text(50)	S (Single value)	M (Mandatory)	
Req_al_CODE	Message code provided to	Text	S (Single value)	M (Mandatory)	

	the message sender in the acknowledgement of the original request for upUIs message. Used for identification of request for aUIs.				
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	null
Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	LBA

Note that:

» Element “Code” cannot be filled in the request message – please send null value

Request sample:

```
{
 "EO_ID": "QCALLA001",
 "Req_al_CODE":"581c9a77-b443-4a2f-92e6-f82bbaa2891b",
 "Code": null,
 "Message_Type": "LBA"
}
```

DESCRIPTION OF THE FIELDS – RESPONSE:

Field	Description	Data Type	Cardinality	Priority	Values
Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	LBA
Error	Indicates the failure of the message reception	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes
Errors	Array containing Error_Code, Error_Descr, InternalId	Text	S (Single value)	M (Mandatory) if Error =1	
Gen_complete	Indicates that the generation of batches is complete	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes
Batches	List of batches generated for unit level UIs request	Text	M (Multiple value)	M (Mandatory) if Gen_complete =1	

Req_no	Identifies the request for unit level UIs to which are the batches associated	Text	S (Single value)	M (Mandatory)	
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	
Checksum	The calculated checksum of the data received	Text	S (Single value)	M (Mandatory)	

The response message contains the hash of the original message in the "Checksum" field.

Checksum = X-OriginalHash sent in the original message - request header.

Successful response sample:
HTTP Status 202


```
{
 "Message_Type": "LBU",
 "Code": "873345b2-882f-4064-91f0-90669b46c30a",
 "Error": 0,
 "Errors": null,
 "Gen_complete": 0,
 "Batches": [],
 "Req_no": "ORD1920000201",
 "Checksum": "G6HF5H"
}
```


Error response:

Set of error responses, that can be sent in this API:

HTTP status	Error Code	Error Description
401	INVALID_OR_EXPIRED_TOKEN	Invalid or expired security token
401	UNAUTHORIZED	Unauthorized call.
400	FAILED_VALIDATION	This error is returned when at least one of the mandatory fields are missing, if an enumeration is wrong
400	INVALID_SIGNATURE	Hash information not matching the message signature.
400	CODE_PROVIDED	Code was provided in context where it's not expected, for example when a manufacturer calls a primary
400	INVALID_MESSAGE_TYPE	When the field "Message_Type" is out of the defined list.
400	INVALID_INPUT_FORMAT	When the body of the message doesn't contain a valid JSON.
400	EIOD_NOT_EXIST_OR_ACTIVE	Check if EOID exists and is active (VAL_ENT_EXIST_EOID, VAL_ENT_ACTIVE_EOID).

400	INVALID_REQUEST_CODE	This error is returned when the sent request code was not found or is not related with sent EO_ID.
500	SYSTEM_ERROR	Internal system error.

Error response sample:

The system provides the sufficient details to allow external systems, administrators to identify the issue precisely in order to act accordingly.

The response message can contain a list of errors

```
"Errors": [
 { << Error >>},
 { << Error >>},
 { << Error >>},
],
```

Each error contains the following information.

- Error_InternalID is the unique identification of the message processing and validation activity.
- Error_Code is the identifier of the type within the systems.
- Error_Descr is the description in human readable format containing specific error information
- Error_Data is closer specification of why the error has occurred

Example of List of errors

```
{
 "Error_InternalID": "257943d2-9437-4eab-bad4-424a24e4ce81",
 "Error_Code": "EOID_NOT_EXIST_OR_ACTIVE",
 "Error_Descr": "EO_ID not exist or active.",
 "Error_Data": "EO_ID or EO_CODE not exist or active."
}
```

Error body sample

```
{
 "Code": "6854f9a6-a2b2-4c08-8000-0173f3c35567",,
 "Message_Type": null,
 "Error": 1,
 "Errors": [
  {
   "Error_InternalID": "257943d2-9437-4eab-bad4-424a24e4ce81",
   "Error_Code":  " FAILED_VALIDATION",
   "Error_Descr": " Not allowed format"
   "Error_Data": "Field 'EO_ID' is mandatory."
  }
 ],
 "Checksum": "G6HF5H"
}
```

GUB – GET UNIT LEVEL UIS FROM BATCH

GENERAL DESCRIPTION

This API is used for gaining all the unit level UIs from a specific batch.

If the request for unit level UIs contains bigger number of UIs than allowed in one batch, the UIs will be divided in more batches. In order to get all UIs from the request for unit level UIs, this API should be called iteratively until you get UIs from all batches belonging to this request. For example if the request contains three batches with generated UIs, this API should be called three times in a row.

One iteration should be: send request and wait for response. After the response is received, next iteration can be done.

To get the list of batches, EOID and batch name returned in response of calling the LBU API must be sent to the request (element “Batch_name”).

DESCRIPTION OF THE FIELDS – REQUEST:

Field	Description	Data Type	Cardinality	Priority	Values
EO_ID	Economic operator identifier code of the submitting entity	Text(50)	S (Single value)	M (Mandatory)	
Batch_Name	Identifier of the batch from which the UIs will be sent	Text	S (Single value)	M (Mandatory)	
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	null
Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	GUB

Note that:

» Element “Code” cannot be filled in the request message – please send null value

Request sample:

```
{
 "EO_ID": "QCALLA001",
 "Batch_Name": "BQCALL0000012r",
 "Code": null,
 "Message_Type": "GUB"
}
```

DESCRIPTION OF THE FIELDS – RESPONSE:

Field	Description	Data Type	Cardinality	Priority	Values
Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	GUB
Error	Indicates the failure of the message reception	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes

Errors	Array containing Error_Code, Error_Descr, InternalId	Text	S (Single value)	M (Mandatory) if Error =1	
upUI	List of unit packet level UIs issued	Text	M (Multiple value)	M (Mandatory)	
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	
Checksum	The calculated checksum of the data received	Text	S (Single value)	M (Mandatory)	

The response message contains the hash of the original message in the "Checksum" field.

Checksum = X-OriginalHash sent in the original message - request header.

Successful response sample:

HTTP Status 202


```
{
 "Message_Type": "GUB",
 "Code": "873345b2-882f-4064-91f0-90669b46c30a",
 "Error": 0,
 "Errors": null,
 "upUI": ["QCALL1jlSp9fel00BK","QCALL1FJQhbPpC00BK","QCALL1MyioFZX200BK"],
 "Checksum": "G6HF5H"
}
```


The structure of unit level UI:

Data element (as per Art. 8.1)	ID Issuer Identifier	Serial Number	Product Code	Time Stamp
Business content	Entity appointed by member state generating the code	Alphanumeric sequence uniquely identifying the pack	Code allowing for the determination of: 1. Place of manufacturing 2. Manufacturing facility 3. Machine 4. Product description 5. Intended market of retail sale 6. Intended shipment route 7. Importer into the Union (when applicable)	Time stamp indicating the date and hour of manufacturing
Format	According to ISO15459-2	Authorized characters set [a-z; A-Z; 0-9] According to ISO646, 62 possibilities per digit	Authorized characters set [a-z; A-Z; 0-9] According to ISO646, 62 possibilities per digit	<YYMMDDhh>
	QCALL			
Length of element	5	9	4	8
	Generated by ID Issuer			Added by Economic operator

Image 3 - Unit level UI structure

The set of examples of unit level UIs:

- Code we supply: QCALL1rg5ryJoy00AY
- Full UI with timestamp: QCALL1rg5ryJoy00AY19032710

Error response:

Set of error responses, that can be sent in this API:

HTTP status	Error Code	Error Description
401	INVALID_OR_EXPIRED_TOKEN	Invalid or expired security token
401	UNAUTHORIZED	Unauthorized call.
400	FAILED_VALIDATION	This error is returned when at least one of the mandatory fields are missing, if an enumeration is wrong
400	INVALID_SIGNATURE	Hash information not matching the message signature.
400	CODE_PROVIDED	Code was provided in context where it's not expected, for example when a manufacturer calls a primary
400	INVALID_MESSAGE_TYPE	When the field "Message_Type" is out of the defined list.
400	INVALID_INPUT_FORMAT	When the body of the message doesn't contain a valid JSON.
400	EIOD_NOT_EXIST_OR_ACTIVE	Check if EOID exists and is active (VAL_ENT_EXIST_EOID, VAL_ENT_ACTIVE_EOID).
400	INVALID_BATCH_NAME	This error is returned when the sent batch name was not found or is not related with sent EO_ID.
500	SYSTEM_ERROR	Internal system error.

Error response sample:

The system provides the sufficient details to allow external systems, administrators to identify the issue precisely in order to act accordingly.

The response message can contain a list of errors

```
"Errors": [
 { << Error >>},
 { << Error >>},
 { << Error >>},
],
```

Each error contains the following information.

- Error_InternalID is the unique identification of the message processing and validation activity.
- Error_Code is the identifier of the type within the systems.
- Error_Descr is the description in human readable format containing specific error information
- Error_Data is closer specification of why the error has occurred

Example of List of errors

```
{
 "Error_InternalID": "257943d2-9437-4eab-bad4-424a24e4ce81",
 "Error_Code": "EOID_NOT_EXIST_OR_ACTIVE",
 "Error_Descr": "EO_ID not exist or active.",
 "Error_Data": "EO_ID or EO_CODE not exist or active."
}
```

Error body sample

```
{
 "Code": "6854f9a6-a2b2-4c08-8000-0173f3c35567",,
 "Message_Type": null,
 "Error": 1,
 "Errors": [
  {
   "Error_InternalID": "257943d2-9437-4eab-bad4-424a24e4ce81",
   "Error_Code":  " FAILED_VALIDATION",
   "Error_Descr": " Not allowed format"
   "Error_Data": "Field 'EO_ID' is mandatory."
  }
 ],
 "Checksum": "G6HF5H"
}
```


GAB – GET AGGREGATED LEVEL UIS FROM BATCH

GENERAL DESCRIPTION

This API is used for gaining all the aggregated level UIs from a specific batch.

If the request for aggregated level UIs contains bigger number of UIs than allowed in one batch, the UIs will be divided in more batches. In order to get all UIs from the request for aggregated level UIs, this API should be called iteratively until you get UIs from all batches belonging to this request. For example if the request contains three batches with generated UIs, this API should be called three times in a row.

One iteration should be: send request and wait for response. After the response is received, next iteration can be done.

To get the list of batches, EOID and batch name returned in response of calling the LBA API must be sent to the request (element “Batch_name”).

DESCRIPTION OF THE FIELDS – REQUEST:

Field	Description	Data Type	Cardinality	Priority	Values
EO_ID	Economic operator identifier code of the submitting entity	Text(50)	S (Single value)	M (Mandatory)	
Batch_Name	Identifier of the batch from which the UIs will be sent	Text	S (Single value)	M (Mandatory)	
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	null
Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	GAB

Note that:

» Element “Code” cannot be filled in the request message – please send null value

Request sample:

```
{
 "EO_ID": "QCALLA001",
 "Batch_Name": "BQCALL000001Ix",
 "Code": null,
 "Message_Type": "GUB"
}
```


DESCRIPTION OF THE FIELDS – RESPONSE:

Field	Description	Data Type	Cardinality	Priority	Values
Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	GAB
Error	Indicates the failure of the message reception	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes
Errors	Array containing Error_Code, Error_Descr, InternalId	Text	S (Single value)	M (Mandatory) if Error =1	
aUI	List of aggregated level UIs issued	Text	M (Multiple value)	M (Mandatory)	
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	
Checksum	The calculated checksum of the data received	Text	S (Single value)	M (Mandatory)	

The response message contains the hash of the original message in the "Checksum" field.

Checksum = X-OriginalHash sent in the original message - request header.

Successful response sample:

HTTP Status 202


```
{
 "Message_Type": "GAB",
 "Code": "873345b2-882f-4064-91f0-90669b46c30a",
 "Error": 0,
 "Errors": null,
 "aUI": ["QCALL1hiOrHkJ2QCALLa00B","QCALL1Lja2Vs2BQCALLa00B"],
 "Checksum": "G6HF5H"
}
```


The structure of aggregated level UI:

Data element (as per Art. 11)	ID Issuer Identifier	Serial Number	Facility Code	Time Stamp
Business content	Entity appointed by member state generating the code	Alphanumeric sequence uniquely identifying the pack	The identifier code of the facility in which the aggregation process took place.	Time stamp indicating the date and hour of aggregation
Format	According to ISO15459-2	Authorized characters set [a-z; A-Z; 0-9] According to ISO646, 62 possibilities per digit	Code issued according to this specification made of: 1. The alphanumeric characters that constitute the ID issuer identification code (5 characters according to ISO15459-2) 2. An alphanumeric sequence which is unique within the code pool of the ID issuer (4 characters [a-z;A-Z;0-9] according to ISO646, 62 possibilities per digit)	<YYMMDDhh>
	QCALL			
Length of element	5	9	9	8
	Generated by ID Issuer			Added by Economic operator

Image 4 – Aggregated level UI structure

The set of examples of aggregated level UIs:

- Code we supply: QCALL1anbJLPWPQCALLa006
- Full UI with timestamp: QCALL1anbJLPWPQCALLa00619032710

Error response:

Set of error responses, that can be sent in this API:

HTTP status	Error Code	Error Description
401	INVALID_OR_EXPIRED_TOKEN	Invalid or expired security token
401	UNAUTHORIZED	Unauthorized call.
400	FAILED_VALIDATION	This error is returned when at least one of the mandatory fields are missing, if an enumeration is wrong
400	INVALID_SIGNATURE	Hash information not matching the message signature.
400	CODE_PROVIDED	Code was provided in context where it's not expected, for example when a manufacturer calls a primary
400	INVALID_MESSAGE_TYPE	When the field "Message_Type" is out of the defined list.
400	INVALID_INPUT_FORMAT	When the body of the message doesn't contain a valid JSON.
400	EIOD_NOT_EXIST_OR_ACTIVE	Check if EOID exists and is active (VAL_ENT_EXIST_EOID, VAL_ENT_ACTIVE_EOID).
400	INVALID_BATCH_NAME	This error is returned when the sent batch name was not found or is not related with sent EO_ID.
500	SYSTEM_ERROR	Internal system error.

Error response sample:

The system provides the sufficient details to allow external systems, administrators to identify the issue precisely in order to act accordingly.

The response message can contain a list of errors

```
"Errors": [
 { << Error >>},
```

```
 { << Error >>},
 { << Error >>},
],
```

Each error contains the following information.

- Error_InternalID is the unique identification of the message processing and validation activity.
- Error_Code is the identifier of the type within the systems.
- Error_Descr is the description in human readable format containing specific error information
- Error_Data is closer specification of why the error has occurred

Example of List of errors

```
{
 "Error_InternalID": "257943d2-9437-4eab-bad4-424a24e4ce81",
 "Error_Code": "EOID_NOT_EXIST_OR_ACTIVE",
 "Error_Descr": "EO_ID not exist or active.",
 "Error_Data": "EO_ID or EO_CODE not exist or active."
}
```

Error body sample

```
{
 "Code": "6854f9a6-a2b2-4c08-8000-0173f3c35567",,
 "Message_Type": null,
 "Error": 1,
 "Errors": [
  {
   "Error_InternalID": "257943d2-9437-4eab-bad4-424a24e4ce81",
   "Error_Code":  " FAILED_VALIDATION",
   "Error_Descr": " Not allowed format"
   "Error_Data": "Field 'EO_ID' is mandatory."
  }
 ],
 "Checksum": "G6HF5H"
}
```

GRD – GET REGISTRATION DATA OF ECONOMIC OPERATOR

GENERAL DESCRIPTION

This API is used for gaining the EOID and EO code of first retail outlet registered by other economic operator using REO API. Submit EOID of the economic operator which registered the first retail outlet.

This API should not be called with more frequency than once per hour (in case the returned information is that the EOID and EO code are not generated yet).

DESCRIPTION OF THE FIELDS – REQUEST:

Field	Description	Data Type	Cardinality	Priority	Values
EO_ID	Economic operator identifier code of the submitting entity	Text(50)	S (Single value)	M (Mandatory)	

	(which requested the REO API)				
Reo_CODE	Message code provided to the message sender in the acknowledgement of the original registration of first retail outlet message. Used for identification of request for registration of first retail outlet.	Text	S (Single value)	M (Mandatory)	
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	null
Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	GRD

Note that:

» Element “Code” cannot be filled in the request message – please send null value

Request sample:

```
{
 "EO_ID": "QCALLA001",
 "Reo_CODE": "0ec1a77b-ffe5-479f-a8d2-722865d29fd9",
 "Code": null,
 "Message_Type": "GRD"
}
```

DESCRIPTION OF THE FIELDS – RESPONSE:

Field	Description	Data Type	Cardinality	Priority	Values
Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	GRD
Error	Indicates the failure of the message reception	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes
Errors	Array containing Error_Code, Error_Descr, InternalId	Text	S (Single value)	M (Mandatory) if Error =1	

Gen_complete	Indicates that the generation of EOID and EO code is complete	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes
EO_ID	Generated EO ID of the registered first retail outlet.	Text	S (Single value)	M (Mandatory) if Gen_complete =1	
EO_CODE	Economic operator's (first retail outlet) confirmation code provided in response to the registration of first retail outlet	Text	S (Single value)	M (Mandatory) if Gen_complete =1	
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	
Checksum	The calculated checksum of the data received	Text	S (Single value)	M (Mandatory)	

The response message contains the hash of the original message in the "Checksum" field.

Checksum = X-OriginalHash sent in the original message - request header.

Successful response sample:
HTTP Status 202


```
{
 "Message_Type": "GRD",
 "Code": "873345b2-882f-4064-91f0-90669b46c30a",
 "Error": 0,
 "Errors": null,
 "Gen_complete": 1,
 "EO_ID": "QCALLXA03",
 "EO_CODE": "d390bd46-83a3-4f84-9380-6cbb2b239c62",
 "Checksum": "G6HF5H"
}
```


The structure of EO identifier code:

The structure of EOID generated from this Id Issuer system is:

- First 5 characters – ID Issuer prefix (QCALL) according to ISO/IEC 15459-2:2015.5
- Another 4 alphanumeric characters – it is possible that the length of this part of EOID may vary in the future

Error response:

Set of error responses, that can be sent in this API:

HTTP status	Error Code	Error Description
401	INVALID_OR_EXPIRED_TOKEN	Invalid or expired security token
401	UNAUTHORIZED	Unauthorized call.

400	FAILED_VALIDATION	This error is returned when at least one of the mandatory fields are missing, if an enumeration is wrong
400	INVALID_SIGNATURE	Hash information not matching the message signature.
400	CODE_PROVIDED	Code was provided in context where it's not expected, for example when a manufacturer calls a primary
400	INVALID_MESSAGE_TYPE	When the field "Message_Type" is out of the defined list.
400	INVALID_INPUT_FORMAT	When the body of the message doesn't contain a valid JSON.
400	EIOD_NOT_EXIST_OR_ACTIVE	Check if EOID exists and is active (VAL_ENT_EXIST_EOID, VAL_ENT_ACTIVE_EOID).
400	INVALID_BATCH_NAME	This error is returned when the sent batch name was not found or is not related with sent EO_ID.
500	SYSTEM_ERROR	Internal system error.

Error response sample:

The system provides the sufficient details to allow external systems, administrators to identify the issue precisely in order to act accordingly.

The response message can contain a list of errors

```
"Errors": [
 { << Error >>},
 { << Error >>},
 { << Error >>},
],
```

Each error contains the following information.

- Error_InternalID is the unique identification of the message processing and validation activity.
- Error_Code is the identifier of the type within the systems.
- Error_Descr is the description in human readable format containing specific error information
- Error_Data is closer specification of why the error has occurred

Example of List of errors

```
{
 "Error_InternalID": "257943d2-9437-4eab-bad4-424a24e4ce81",
 "Error_Code": "EOID_NOT_EXIST_OR_ACTIVE",
 "Error_Descr": "EO_ID not exist or active.",
 "Error_Data": "EO_ID or EO_CODE not exist or active."
}
```

Error body sample

```
{
 "Code": "6854f9a6-a2b2-4c08-8000-0173f3c35567",,
 "Message_Type": null,
 "Error": 1,
 "Errors": [
```

```
  {
   "Error_InternalID": "257943d2-9437-4eab-bad4-424a24e4ce81",
   "Error_Code":  " FAILED_VALIDATION",
   "Error_Descr": " Not allowed format"
   "Error_Data": "Field 'EO_ID' is mandatory."
  }
 ],
 "Checksum": "G6HF5H"
}
```


REQUIRED INTERFACE

This chapter contains details of all interfaces required by Id Issuer system from systems of economic operators.

We expect, that in the required APIs will also be used the OAuth 2 to authorize access implemented the same way like described in chapter "Authentication".

Each type of message (API) can have a separate endpoint, or there can be one endpoint for all required APIs. To all this endpoints we require access using HTTPS protocol. We expect that the certificate will be issued by some trusted certificate authority.

Communication between the Id Issuer and interacting participants of the tobacco industry is secured by TSL 1.2 encryption AES256 cypher. Cypher suites that are less secure are not supported.

IRU – MESSAGE TO REPORT THE ISSUANCE OF SERIAL NUMBERS AT UNIT PACKET LEVEL

GENERAL DESCRIPTION

Request for reporting the issuance of serial numbers at unit packet level. Because of the manufacturer's demands to have an provided interface, where they will be able to download the generated unit level UIs, we are considering the future support of this API.

DESCRIPTION OF THE FIELDS – REQUEST:

Field	Description	Data Type	Cardinality	Priority	Values
EO_ID	Economic operator identifier code of the submitting entity (either EU manufacturer or EU importer)	Text(50)	S (Single value)	M (Mandatory)	
Event_Time	Intended time of event occurrence	Time(s)	S (Single value)	M (Mandatory)	
F_ID	Facility identifier code	Text(50)	S (Single value)	M (Mandatory)	
Process_Type	Indication if the production process involves machinery	Boolean	S (Single value)	M (Mandatory)	0 – No (only for fully hand made products) 1 – Yes
M_ID	Machine identifier code	Text(50)	S (Single value)	M (Mandatory)	

P_Type	Type of tobacco product	Integer	S (Single value)	M (Mandatory)	Value from Tabacco Product Type
P_OtherType	Description of other type of tobacco product	Text	S (Single value)	M (Mandatory)	
P_Brand	Brand of tobacco product	Text	S (Single value)	M (Mandatory)	
P_Weight	Average gross weight of unit packet, including packaging, in grams with 0,1 gram accuracy	Decimal	S (Single value)	M (Mandatory)	
TP_ID	The identification number of the product used in the EU-CEG system.	Text(14)	S (Single value)	M (Mandatory), if Intended_ Market is an EU country	
TP_PN	Tobacco product number used in the EU-CEG system	Text(30)	S (Single value)	M (Mandatory), if Intended_ Market is an EU country	
Intended_Market	Intended country of retail sale	Text(2)	S (Single value)	M (Mandatory)	
Intended_Route1	Indication if the product is intended to be moved across country boarders with terrestrial transport.	Boolean	S (Single value)	M (Mandatory)	0 – No 1 - Yes
Intended_Route2	The first country of terrestrial transport after the product leaves the Member State of manufacturing or the Member State of importation.	Text(2)	S (Single value)	M (Mandatory)	
Import	ndication if the product is imported into the EU	Boolean	S (Single value)	M (Mandatory)	0 – No 1 - Yes
Req_Quantity	Requested quantity of unit packet level UIs	Integer	S (Single value)	M (Mandatory)	
upUI	List of unit packet level UIs issued		M (Multiple value)	M (Mandatory)	
Message_type	The identifier of the type of	Text	S (Single value)	M (Mandatory)	IRU

	message that the response refers to.				
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	

Request sample:


```
{
 "EO_ID":" QCUKR+1AB020054",
 "Event_Time" : "2018-08-23T07:32:20.7878086+00:00",
 "F_ID": "QCUKR<1AB020054000049",
 "Process_Type":0,
 "M_ID":"Machine Id A",
 "P_Type":2,
 "P_OtherType":null,
 "P_Brand":"Product brand A",
 "P_Weight":10.0,
 "TP_ID":" 1234",
 "TP_PN":"1234",
 "Intended_Market":"BG",
 "Intended_Route1":1,
 "Intended_Route2":"BG",
 "Import":0,
 "Req_Quantity":2,
 "upUI":[" QCALL1rg5ryJoy00AY"," QCALL1rg5ryJoy00BY"],
 "Message_Type": "IRU"
 "Code": "873345b2-882f-4064-91f0-90669b46c30a"
}
```


The structure of unit level UI:

Data element (as per Art. 8.1)	ID Issuer Identifier	Serial Number	Product Code	Time Stamp
Business content	Entity appointed by member state generating the code	Alphanumeric sequence uniquely identifying the pack	Code allowing for the determination of: 1. Place of manufacturing 2. Manufacturing facility 3. Machine 4. Product description 5. Intended market of retail sale 6. Intended shipment route 7. Importer into the Union (when applicable)	Time stamp indicating the date and hour of manufacturing
Format	According to ISO15459-2	Authorized characters set [a-z; A-Z; 0-9] According to ISO646, 62 possibilities per digit	Authorized characters set [a-z; A-Z; 0-9] According to ISO646, 62 possibilities per digit	<YYMMDDhh>
	QCALL			
Length of element	5	9	4	8
	Generated by ID Issuer			Added by Economic operator

Image 5 - Unit level UI structure

The set of examples of unit level UIs:

- Code we supply: QCALL1rg5ryJoy00AY
- Full Uis with timestamp: QCALL1rg5ryJoy00AY19032710

DESCRIPTION OF THE FIELDS – RESPONSE:

Field	Description	Data Type	Cardinality	Priority	Values
Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	IRU
Error	Indicates the failure of the message reception	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes
Errors	Array containing Error_Code, Error_Descr, InternalId	Text	S (Single value)	M (Mandatory) if Error =1	
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	
Checksum	The calculated checksum of the data received	Text	S (Single value)	M (Mandatory)	

MD5 hash function is used: Checksum = MD5(message body)

The response message contains the hash of the original message in the "Checksum" field.

Successful response sample:
HTTP Status 202

```
{
 "Code": "873345b2-882f-4064-91f0-90669b46c30a",
 "Message_Type": "IRU",
 "Error": 0,
 "Errors": null,
 "Checksum": "G6HF5H"
}
```

Error response:

HTTP status	Error Code	Error Description
401	SECURITY_INVALID_TOKEN	Invalid security token
401	SECURITY_EXPIRED_TOKEN	Expired security token
400	FAILED_VALIDATION	This error is returned when at least one of the mandatory fields are missing, if an enumeration is wrong
400	REQUIRED_FIELD_FAILED_VALIDATION	Mandatory fields is missing
400	MAX_LENGTH_ FAILED_VALIDATION	Over max length of field.
400	INVALID_REQUEST_FORMAT	This error is returned when at least one of the mandatory fields are missing.
400	INVALID_MESSAGE_TYPE	When the field “Message_Type” is out of the defined list.

400	INVALID_INPUT_FORMAT	When the body of the message doesn't contain a valid JSON.
400	EIOD_NOT_EXIST_OR_ACTIVE	Check if EOID, exists and is active (VAL_ENT_EXIST_EOID, VAL_ENT_ACTIVE_EOID)
400	FID_NOT_EXIST_OR_ACTIVE	Check if FID, exists and is active (VAL_ENT_EXIST_FID, VAL_ENT_ACTIVE_FID)
400	FID_NOT_RELATED_TO_EOID	Check if EOID FID relation (VAL_ENT_REL_EOID_FID)
400	MID_NOT_EXIST_OR_ACTIVE	Check if MID, exists and is active (VAL_ENT_EXIST_MID, VAL_ENT_ACTIVE_MID)
400	MID_NOT_RELATED_TO_FID	Check if FID MID relation (VAL_ENT_REL_FID_MID)
500	SYSTEM_ERROR	Internal system error.

Error response sample:

The system should provide the sufficient details to allow external systems, administrators to identify precisely the issue in order to act accordingly.

The response message can contain a list of error

```
"Errors": [
 { << Error >>},
 { << Error >>},
 { << Error >>},
],
```

Each error contains the following information.

- Error_InternalID is the unique identification of the message processing and validation activity.
- Error_Code is the identifier of the type within the systems.
- Error_Descr is the description in human readable format containing specific error information
- Error_Data is the data for which the error is talking about. This can be used for EO_IDs, F_IDs, M_IDs and UIs

Example of List of errors

```
{
 "Error_InternalID": "yndkFz7TBEO706frD38hzA",
 "Error_Code": "EOID_NOT_EXIST_OR_ACTIVE",
 "Error_Descr": "The EconomicOperatorIdentifier field is unknown.",
 "Error_Data": "QCALL1234"
}
```

Error body sample

```
{
 "Code": " 6854f9a6-a2b2-4c08-8000-0173f3c35567",,
 "Message_Type": null,
 "Error": 1,
 "Errors": [
  {
   "Error_InternalID": "yndkFz7TBEO706frD38hzA",
   "Error_Code":  "INVALID_REQUEST_FORMAT",
```

```
   "Error_Descr": "The EconomicOperatorIdentifier field is required."
   "Error_Data": null
  }
 ],
 "Checksum": "G6HF5H"
}
```


IRA – REQUEST FOR REPORTING THE ISSUANCE OF SERIAL NUMBERS AT AGGREGATED LEVEL

GENERAL DESCRIPTION

Request for reporting the issuance of serial numbers at aggregated level. Because of the manufacturer's demands to have an provided interface, where they will be able to download the generated aggregated level UIs, we are considering the future support of this API.

DESCRIPTION OF THE FIELDS – REQUEST:

Field	Description	Data Type	Cardinality	Priority	Values
EO_ID	Economic operator identifier code of the submitting entity (either EU manufacturer or EU importer)	Text(50)	S (Single value)	M (Mandatory)	
Event_Time	Intended time of event occurrence	Time(s)	S (Single value)	M (Mandatory)	
F_ID	Facility identifier code	Text(50)	S (Single value)	M (Mandatory)	
Req_Quantity	Requested quantity of unit packet level UIs	Integer	S (Single value)	M (Mandatory)	
aUI	Aggregated level unique identifier coded	Text(100)	M (Multiple value)	O (Optional)	
Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	IRA
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	

Request sample:


```
{
 "EO_ID": "QCUKR+1AB020054",
 "Event_Time" : "2018-08-23T07:32:20.7878086+00:00",
 "F_ID": "QCUKR<1AB020054000049",
 "Req_Quantity": 2,
 "aUI": [ " QCALL1anbJLPWPQCALLa006",
 " QCALL1anbJLPWPQCALLa007" ],
 "Message_Type": "IRA",
```

```
 "Code": "873345b2-882f-4064-91f0-90669b46c30a"
}
```


The structure of unit level UI:

Data element (as per Art. 11)	ID Issuer Identifier	Serial Number	Facility Code	Time Stamp
Business content	Entity appointed by member state generating the code	Alphanumeric sequence uniquely identifying the pack	The identifier code of the facility in which the aggregation process took place.	Time stamp indicating the date and hour of aggregation
Format	According to ISO15459-2	Authorized characters set [a-z; A-Z; 0-9] According to ISO646, 62 possibilities per digit	Code issued according to this specification made of: 1. The alphanumeric characters that constitute the ID issuer identification code (5 characters according to ISO15459-2) 2. An alphanumeric sequence which is unique within the code pool of the ID issuer (4 characters [a-z;A-Z;0-9] according to ISO646, 62 possibilities per digit)	<YYMMDDhh>
	QCALL			
Length of element	5	9	9	8
	Generated by ID Issuer			Added by Economic operator

Image 6 – Aggregated level UI structure

The set of examples of aggregated level UIs:

- Code we supply: QCALL1anbJLPWPQCALLa006
- Full Uis with timestamp: QCALL1anbJLPWPQCALLa00619032710

DESCRIPTION OF THE FIELDS – RESPONSE:

Field	Description	Data Type	Cardinality	Priority	Values
Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	IRA
Error	Indicates the failure of the message reception	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes
Errors	Array containing Error_Code, Error_Descr, InternalId	Text	S (Single value)	M (Mandatory) if Error =1	
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	
Checksum	The calculated checksum of the data received	Text	S (Single value)	M (Mandatory)	

MD5 hash function is used: Checksum = MD5(message body)

The response message contains the hash of the original message in the "Checksum" field.

Successful response sample:
HTTP Status 202

```
{
 "Code": "873345b2-882f-4064-91f0-90669b46c30a",
 "Message_Type": "IRA",
 "Error": 0,
 "Errors": null,
 "Checksum": "G6HF5H"
}
```

Error response:

HTTP status	Error Code	Error Description
401	SECURITY_INVALID_TOKEN	Invalid security token
401	SECURITY_EXPIRED_TOKEN	Expired security token
400	FAILED_VALIDATION	This error is returned when at least one of the mandatory fields are missing, if an enumeration is wrong
400	REQUIRED_FIELD_FAILED_VALIDATION	Mandatory fields is missing
400	MAX_LENGTH_ FAILED_VALIDATION	Over max length of field.
400	INVALID_REQUEST_FORMAT	This error is returned when at least one of the mandatory fields are missing.
400	INVALID_MESSAGE_TYPE	When the field “Message_Type” is out of the defined list.
400	INVALID_INPUT_FORMAT	When the body of the message doesn’t contain a valid JSON.
400	EIOD_NOT_EXIST_OR_ACTIVE	Check if EOID, exists and is active (VAL_ENT_EXIST_EOID, VAL_ENT_ACTIVE_EOID)
400	FID_NOT_EXIST_OR_ACTIVE	Check if FID, exists and is active (VAL_ENT_EXIST_FID, VAL_ENT_ACTIVE_FID)
400	FID_NOT_RELATED_TO_EOID	Check if EOID FID relation (VAL_ENT_REL_EOID_FID)
500	SYSTEM_ERROR	Internal system error.

Error response sample:

The system should provide the sufficient details to allow external systems, administrators to identify precisely the issue in order to act accordingly.

The response message can contain a list of error

```
"Errors": [
 { << Error >>},
 { << Error >>},
 { << Error >>},
],
```

Each error contains the following information.

- Error_InternalID is the unique identification of the message processing and validation activity.
- Error_Code is the identifier of the type within the systems.

- Error_Descr is the description in human readable format containing specific error information
- Error_Data is the data for which the error is talking about. This can be used for EO_IDs, F_IDs, M_IDs and UIs

Example of List of errors

```
{
 "Error_InternalID": "yndkFz7TBEO706frD38hzA",
 "Error_Code": "EOID_NOT_EXIST_OR_ACTIVE",
 "Error_Descr": "The EconomicOperatorIdentifier field is unknown.",
 "Error_Data": "QCALL1234"
}
```

Error body sample

```
{
 "Code": " 6854f9a6-a2b2-4c08-8000-0173f3c35567",,
 "Message_Type": null,
 "Error": 1,
 "Errors": [
  {
   "Error_InternalID": "yndkFz7TBEO706frD38hzA",
   "Error_Code":  "INVALID_REQUEST_FORMAT",
   "Error_Descr": "The EconomicOperatorIdentifier field is required."
   "Error_Data": null
  }
 ],
 "Checksum": "G6HF5H"
}
```

REOD - DATA REGISTRATION OF AN ECONOMIC OPERATOR

GENERAL DESCRIPTON

The REOD message is the response to the REO message. This message can be issued in an asynchronous manner and contains the EO_ID. Because of the manufacturer's demands to have an provided interface, where they will be able to download the generated EOID, we are considering the future support of this API.

DESCRIPTION OF THE FIELDS – REQUEST:

Field	Description	Data Type	Cardinality	Priority	Values
EO_ID	Economic operator identifier code of the submitting entity (either EU manufacturer or EU importer)	Text(50)	S (Single value)	M (Mandatory)	
EO_Name1	Economic operator's registered name	Text	S (Single value)	M (Mandatory)	
EO_Name2	Economic operator's alternative or abridged name	Text	S (Single value)	O (Optional)	

EO_Address	Economic operator's address – street name, house number, postal code, city	Text	S (Single value)	M (Mandatory)	
EO_CountryReg	Economic operator's country of registration	Text(2)	S (Single value)	M (Mandatory)	Code from Country code list
EO_Email	Economic operator's email address; used to inform about registration process, incl. subsequent changes and other required correspondence	Text	S (Single value)	M (Mandatory)	
VAT_R	Indication of the VAT registration status	Boolean	S (Single value)	M (Mandatory)	0 – No VAT registration 1 – VAT number exist
VAT_N	Economic operator's VAT number	Text	S (Single value)	M (Mandatory), if VAT_R = 1	
TAX_N	Economic operator's tax registration number	Text	S (Single value)	M (Mandatory), if VAT_R = 0	
EO_ExciseNumber1	Indication if the economic operator has an excise number issued by the competent authority for the purpose of identification of persons/premises	Boolean	S (Single value)	M (Mandatory)	0 – No SEED number 1 – SEED number exists
EO_ExciseNumber2	Economic operator's excise number issued by the competent authority for the purpose of identification of persons/premises	Text(13)	S (Single value)	M (Mandatory), if EO_Excise Number1 = 1	
OtherEOID_R	Indication if the economic operator has been allocated	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes

	an identifier by another ID Issuer				
OtherEOID_N	Economic operator identifier codes allocated by other ID Issuers	Text(50)	M (Multiple values)	M (Mandatory), if OtherEOID_R = 1	
Reg_3RD	Indication if the registration is made on behalf of a retail outlet operator not otherwise involved in the tobacco trade	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes
Reg_EOID	Identifier of the economic operator that acts on behalf of a retail outlet operator not otherwise involved in the tobacco trade	Text(50)	S (Single value)	M (Mandatory), if Reg_3RD = 1	
EO_OtherID	Optional identifier	Text(50)	S (Single value)	M (Mandatory)	
Message_Type	The identifier of the type of message.	Text	S (Single value)	M (Mandatory)	REOD
Code	Unique identifier of the message. Used for recall too	Text	S (Single value)	M (Mandatory)	

Request sample:


```
{
 "EO_ID": " QCUKR+1AB020054",
 "EO_CODE": "123",
 "EO_Name1": "Example Legal Entity",
 "EO_Name2": "",
 "EO_Address": "59 Legal Street",
 "EO_CountryReg": "DE" ,
 "EO_Email": "email@test.com",
 "VAT_R": 1,
 "VAT_N": "VATNumber 1",
 "TAX_N": "Tax",
 "EO_ExciseNumber1":1,
 "EO_ExciseNumber2": "LA111FD",
 "OtherEOID_R": 1,
 "OtherEOID_N": [ "" ],
 "Reg_3RD": 0,
 "Reg_EOID": "",
```

```
 "EO_OtherID ": "1234567890128",
 "Message_Type": "REOD",
 "Code": " 6854f9a6-a2b2-4c08-8000-0173f3c35567"
}
```

DESCRIPTION OF THE FIELDS – RESPONSE:

Field	Description	Data Type	Cardinality	Priority	Values
Message_type	The identifier of the type of message that the response refers to.	Text	S (Single value)	M (Mandatory)	REOD
Error	Indicates the failure of the message reception	Boolean	S (Single value)	M (Mandatory)	0 – No 1 – Yes
Errors	Array containing Error_Code, Error_Descr, InternalId	Text	S (Single value)	M (Mandatory) if Error =1	
Code	Unique identifier of the message. Used for recall too.	Text	S (Single value)	M (Mandatory)	
Checksum	The calculated checksum of the data received	Text	S (Single value)	M (Mandatory)	

MD5 hash function is used: Checksum = MD5(message body)

The response message contains the hash of the original message in the "Checksum" field.

Successful response sample:
HTTP Status 202

```
{
 "Code": "873345b2-882f-4064-91f0-90669b46c30a",
 "Message_Type": "REOD",
 "Error": 0,
 "Errors": null,
 "Checksum": "G6HF5H"
}
```

Error response:

HTTP status	Error Code	Error Description
401	SECURITY_INVALID_TOKEN	Invalid security token
401	SECURITY_EXPIRED_TOKEN	Expired security token
400	FAILED_VALIDATION	This error is returned when at least one of the mandatory fields are missing, if an enumeration is wrong
400	REQUIRED_FIELD_FAILED_VALIDATION	Mandatory fields is missing

400	MAX_LENGTH_ FAILED_VALIDATION	Over max length of field.
400	INVALID_REQUEST_FORMAT	This error is returned when at least one of the mandatory fields are missing.
400	INVALID_MESSAGE_TYPE	When the field “Message_Type” is out of the defined list.
400	INVALID_INPUT_FORMAT	When the body of the message doesn’t contain a valid JSON.
400	EXCISE_NUMBER_NOT_VALID	SEED type format wrong.
500	SYSTEM_ERROR	Internal system error.

Error response sample:

The system should provide the sufficient details to allow external systems, administrators to identify precisely the issue in order to act accordingly.

The response message can contain a list of error

```
"Errors": [
 { << Error >>},
 { << Error >>},
 { << Error >>},
],
```

Each error contains the following information.

- Error_InternalID is the unique identification of the message processing and validation activity.
- Error_Code is the identifier of the type within the systems.
- Error_Descr is the description in human readable format containing specific error information
- Error_Data is the data for which the error is talking about. This can be used for EO_IDs, F_IDs, M_IDs and UIs

Example of List of errors

```
{
 "Error_InternalID": "yndkFz7TBEO706frD38hzA",
 "Error_Code": "EOID_NOT_EXIST_OR_ACTIVE",
 "Error_Descr": "The EconomicOperatorIdentifier field is unknown.",
 "Error_Data": "QCALL1234"
}
```

Error body sample

```
{
 "Code": " 6854f9a6-a2b2-4c08-8000-0173f3c35567",,
 "Message_Type": null,
 "Error": 1,
 "Errors": [
  {
   "Error_InternalID": "yndkFz7TBEO706frD38hzA",
   "Error_Code":  "INVALID_REQUEST_FORMAT",
   "Error_Descr": "The EconomicOperatorIdentifier field is required."
   "Error_Data": null
  }
 ],
 "Checksum": "G6HF5H"
}
```

ENDPOINTS

	URL
The authentication endpoint	https://< ENV >.< BASE_URL >/api/token
The Id Issuer to economic operator endpoint	https://< ENV >.< BASE_URL >/api/ii2eo

< BASE_URL > : base url

< ENV > : environment name

BASE URL:

- Ireland: idissuer.ie

ENVIROMENT NAME:

- Test environment: tapp
- Production environment: app