GI ID ISSUER SPECIFICATIONS

V1.0

dentsu
TRACKING

Summary of changes

Date	Version	Done by	Comment
10.03.2026	1.0	Dentsu Tracking	Initial version

Publication

Date	Version	Submitted to
13.03.2026	1.0	Publicly available

Table of contents

1 Introduction

This document presents a technical user guide on the functioning of the GI ID Issuer services operated by Dentsu Tracking in the context of the GI Track & Trace system.

This guide covers all relevant topics related to the ID Issuer services, including onboarding and user management, identifier codes and unique identifiers, credentials and validation rules. Readers also have access to technical details necessary for Economic Operators to manage their registry data.

1.1 Definitions and abbreviations

Expression or Acronym	Definition
CIR 2018/574	Commission Implementing Regulation (EU) 2018/574 as amended by Commission Implementing Regulation (EU) 2023/448, as applied and amended by The Tobacco Products (Traceability System and Security Features) (Amendments) (EU Exit) Regulations 2020
EO	Economic Operator
EOID	Economic Operator Identifier code
FID	Facility Identifier Code
MID	Machine Identifier Code
FCTC Protocol	The Protocol to Eliminate Illicit Trade in Tobacco Products is the first protocol to the WHO Framework Convention on Tobacco Control (WHO FCTC)
HMGoG	His Majesty's Government of Gibraltar 'the Authority' (the Contracting Authority)
KPI	Key Performance Indicator
RoW	Rest of World. Other countries outside Gibraltar.
SLA	Service Level Agreement
UI	Unique Identifier. The alphanumeric code enabling the identification of a unit packet or an aggregated packaging of tobacco products.
GI	Gibraltar

Service Provider	Third party acting on behalf of one or multiple Economic Operator, submitting messages to the GI Gateway
ATD	Anti-Tampering Devices.

2 System Overview

This document defines the user guide and technical specifications for the ID Issuer of the GI Tobacco Track & Trace System established and operated by Dentsu Tracking. The information provided in this document includes information about the code format, the process to onboard into the system and request credentials, the support process and a detailed specification of the API and the methods available to Economic Operators.

2.1 Key design principles

The GI Track & Trace system consists of the following sub-systems:

- **ID issuer:** generates and issues unique identifier codes for tobacco products destined for and travelling through the GI Territory (Gibraltar), as well as identifier codes that enable the registration of Economic Operator, Facilities and Machines in the system. This document focuses on this component.
- **GI Gateway:** Economic operators must record all product movements and transactional data for tobacco products, from the manufacturer to last economic operator before the first retail outlet. All recorded data must be transmitted to the GI Gateway which will perform the applicable business and technical validations on the submitted data and provide a positive or negative acknowledgment message to the sender accordingly. For more information about the GI Gateway please refer to the "GI Gateway technical specifications" document.
- **GI Data Repository:** The received data is subsequently transferred to the GI Data Repository where it is accessible to authorised personnel of HMGoG and any other nominated authority.

2.2 GI Track and Trace System

The following diagram represents the overall system design of the GI Track & Trace System.

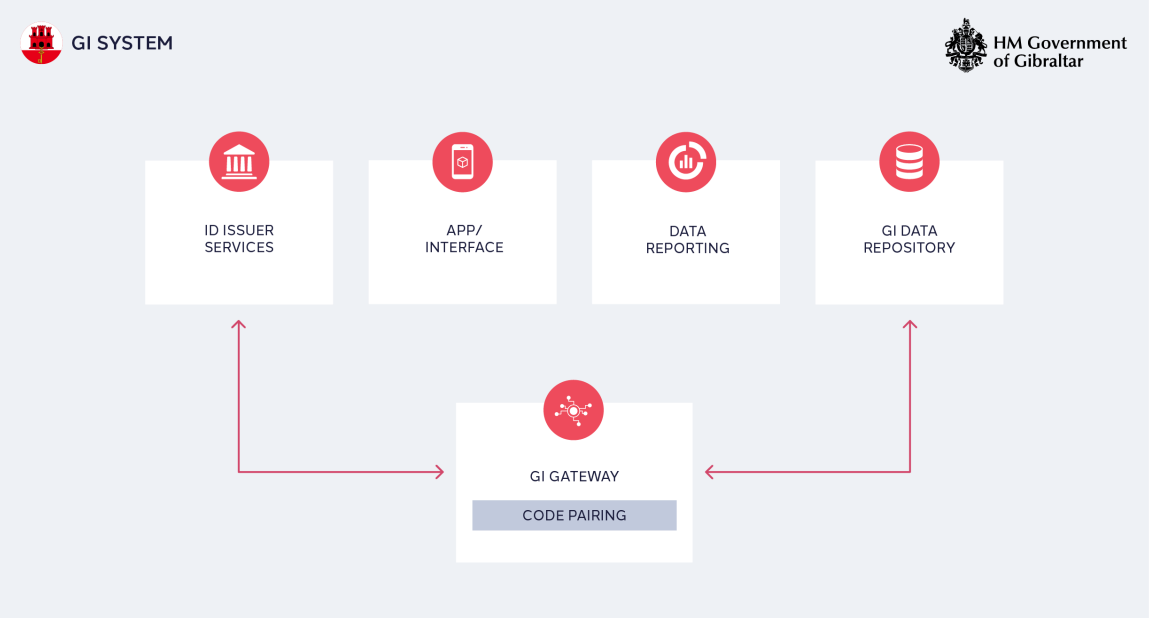

Figure 1 Overall system interconnection

2.2.1 GI ID Issuer

Tobacco manufacturers and importers, along with all other businesses in the tobacco supply chain, must be registered with the Dentsu GI ID Issuer to obtain an economic operator ID (EOID). Once registered, manufacturers must also register for Machine IDs (MID) for machines integral to the manufacturing process. All businesses in the tobacco supply chain must register for Facility IDs (FID) for all premises where they manufacture, store, handle or sell their products. Registered IDs (either EOIDs, MIDs or FIDs) are provided free of charge by Dentsu.

The process to request and/or modify additional identifier codes (EOID, FID, MID) is described in this document.

Each unit packet of tobacco products (cigarettes, hand rolling tobacco, and other tobacco products) manufactured in or imported into GI, must have a GI UI applied to and/or associated with the individual product.

Where tobacco products imported into GI are already marked with a UI issued by another country operating a track and trace system, such as in the EU, the GI UI is not printed on the packet. Instead, it is digitally associated with the UI already marked on the packet. This is a mandatory requirement in GI legislation.

UIs come in the form of an alphanumeric code. They can be either unit packet UIs or aggregate UIs. An aggregate UI is used for anything that holds more than one unit packet e.g., cartons, master cases, pallets, trailers and shipping containers. It is then linked to the unit packets that it contains. Manufacturers and importers can request unit packet UIs from Dentsu for a fee. Aggregate UIs can be either purchased from Dentsu or generated by registered economic operators themselves. Economic operators are responsible for encoding the digital GI UI and ensuring its correct application.

When Economic Operators decide to self-generate the aggregated level identifier codes, these must comply with the ISOs described in the CIR (EU) 2018/574.

Figure 2 – GI Track&Trace System - ID Issuer services

2.2.2 GI Gateway

All data regarding cigarettes, hand rolling tobacco and other tobacco products manufactured in and imported into Gibraltar are routed to the GI data repository.

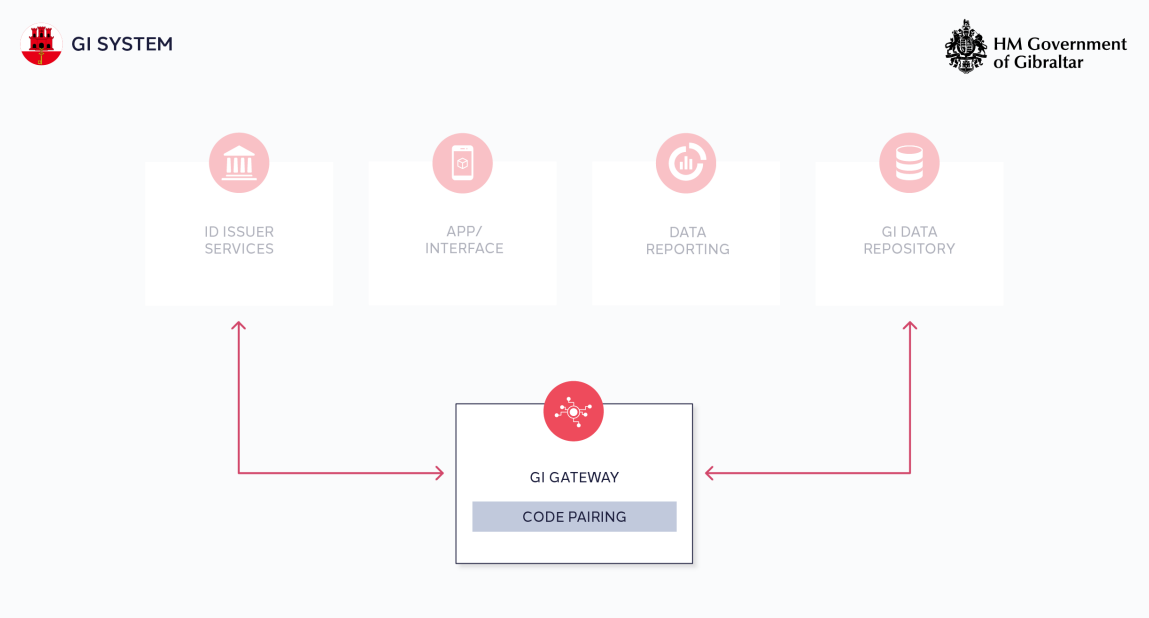

Figure 3 – GI Track&Trace System - GI Gateway

The GI Gateway is a central component of the GI track and trace system. The GI Gateway receives events from the GI ID Issuer and messages reported by Economic Operators and forwards them to the GI Repository. It therefore serves as a central reporting point for all Economic Operators. The Gateway also ensures a high level of data reporting quality and the overall integrity of all reported UIs by the means of sophisticated, high performing validation mechanisms (business and technical validations) that are applied to all received messages.

The following diagram shows all parties and components involved in this process:

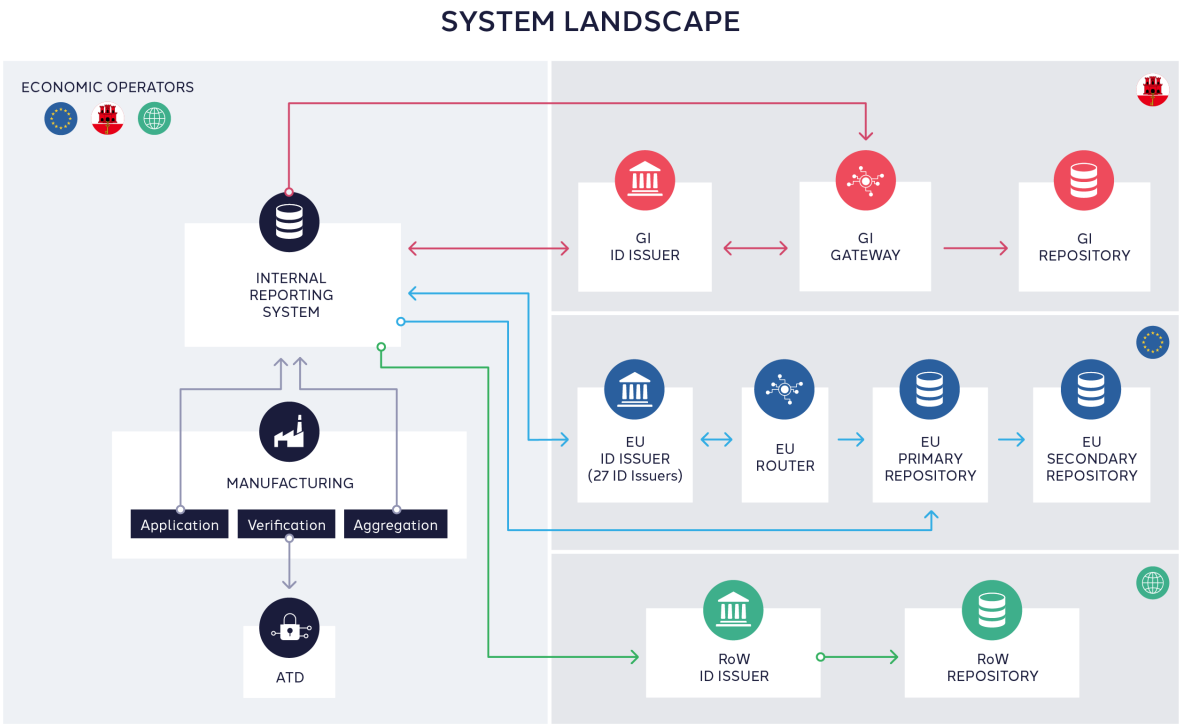

Figure 4 - GI Track & Trace involved parties

Economic Operators are responsible for transmitting messages to the GI Gateway on the reporting of product movement and transactional events.

The GI Gateway only transmits data to the GI Repository and is not interfacing with any other Track & Trace system from third country regimes, such as the EU tobacco traceability system. Economic Operators are responsible for ensuring transmission of relevant data to any other third country systems, such as the EU system, where applicable.

2.2.3 GI Data Repository

All information registered in and transferred to the GI track and trace system is routed by the GI Gateway to the GI Data Repository, where such information is made accessible to the GI Authority. All data is stored in a secure environment that is only accessible to authorised government personnel.

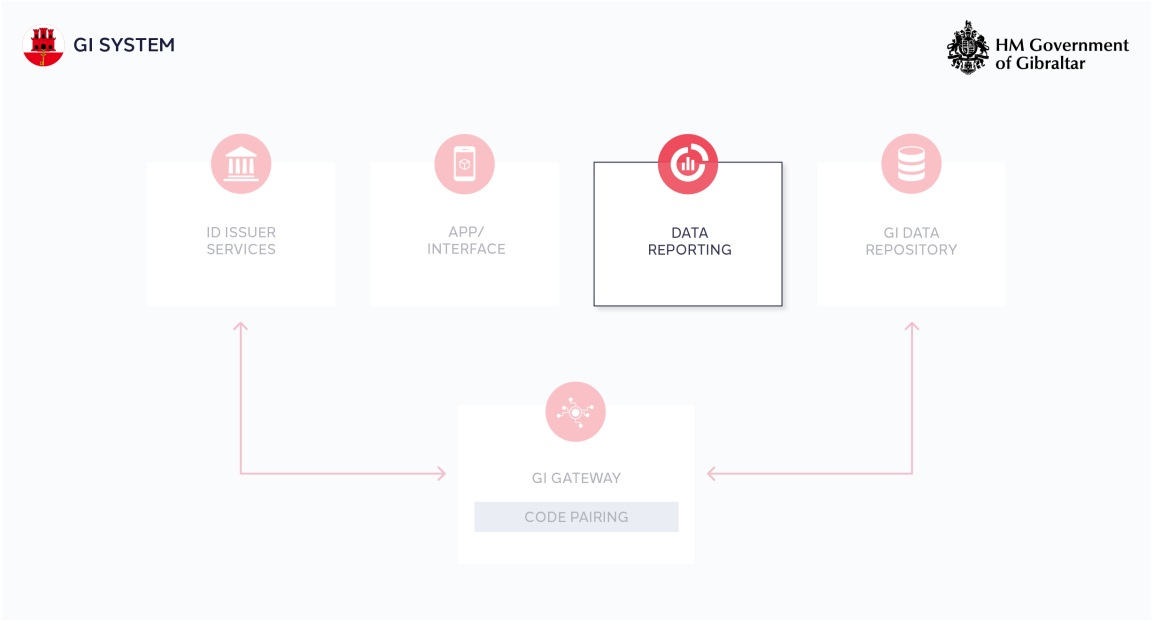

Figure 5 - GI Data Repository & Reporting

3 Onboarding process

New businesses requiring access to the system can go through a self-registration process to onboard themselves.

Economic Operators must request credentials to access the ID Issuer portal. The account creation for an Economic Operator requires submission of pre-defined information. This information allows the identification of the organisation. If economic operators intend to request UIs, additional billing information must be completed.

Once onboarded, each Economic Operator will have 1 Organisation created into the GI ID Issuer as their "Organisation Account". Via this account, a company can then register additional EO-IDs.

The ID Issuer Portal can be accessed via the GI Tobacco Track & Trace Project website:

https://www.gitobaccotracing.com/

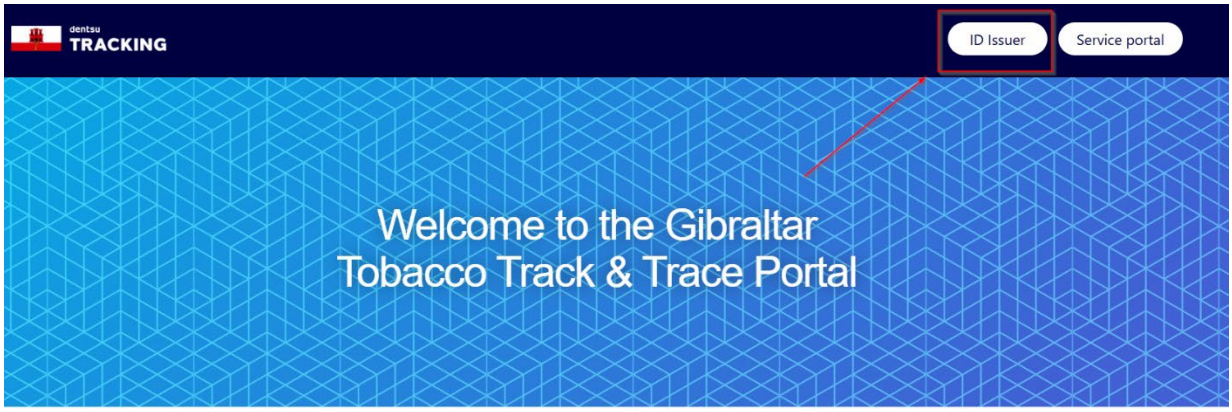

Figure 6 - Dentsu GI Document Center

The onboarding process is initiated via the [**ID ISSUER**] button at the top right of the main page of the GI Track&Trace System webpage, then accessing registration through the "Register" option.

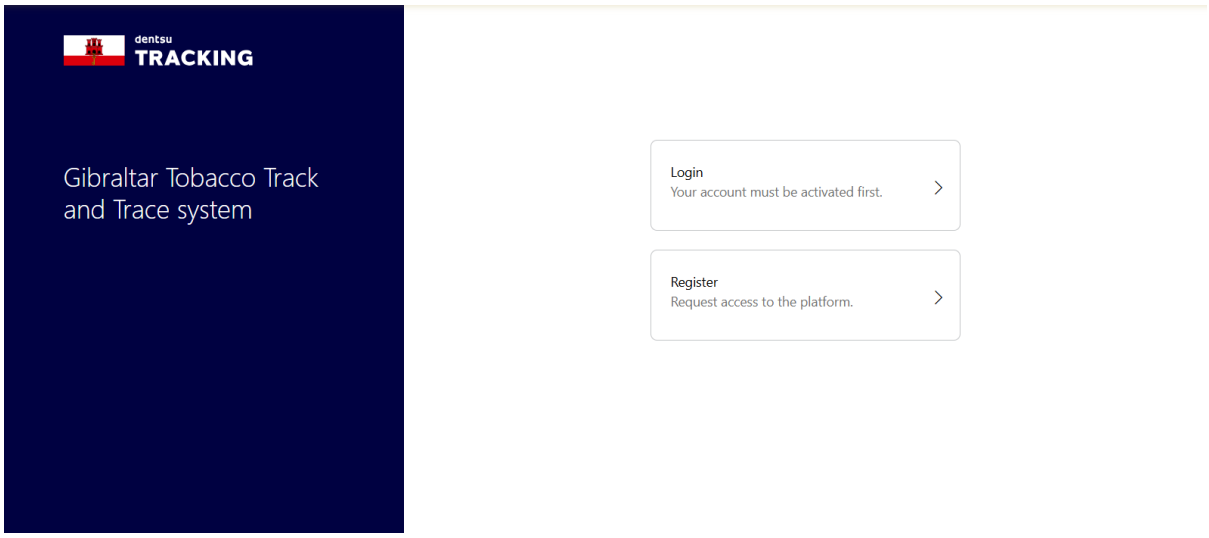

Figure 7 – Onboarding Wizard

Account creation consists of four user friendly steps:

1) **Step 1** – Select the Organisation Type (related to the type of the business performed) Manufacturer/Importer, Other Economic Operator or Service Provider

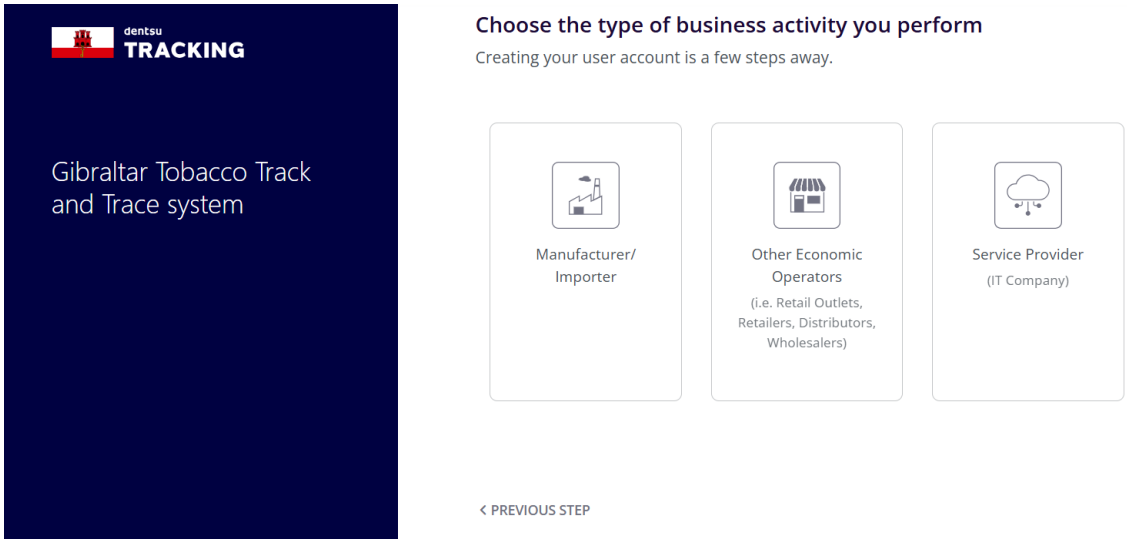

Figure 8 – Onboarding Wizard – Organisation Type

Manufacturer/Importer can perform the following activities:
- request upUIs (unit level unique identifiers) and aUIs (aggregated level unique identifiers).
- manage their registry by requesting/modifying/de-registering Identifier Codes (EOIDs, FIDs, MIDs).
- request API credentials for the GI Track & Trace System.

Other Economic Operator can perform the following activities:
- request aUIs (aggregated level unique identifiers) (NOTE: Only Other Economic Operators that have provided billing information can request unique identifiers. If your organisation is already onboarded and wish to order aggregated level UIs, please contact gi-tnt-support@dentsu.com).
- manage their registry by requesting/modifying/de-registering Identifier Codes (EOIDs, FIDs, MIDs).
- request API credentials for the GI Track & Trace System.

Service Provider can perform the following activities:
- request API credentials for the GI Track & Trace System to submit messages to the GI Gateway on behalf of customers.

2) **Step 2** – Fill in the Organisation details that are to be registered in the GI System (the following set of fields are common for all Organisation Types):

    - Organisation Name
    - Street
    - Number
    - City
    - Postal Code
    - Country
    - Relevant taxation identifier

3) Step 3 – Select whether or not your company requires to order unique-level unique identifiers and/or aggregated level unique identifiers from the GI ID Issuer via the "I wish to order UIDs" or" I wish to order aggregated level UIs" checkbox.

NOTE that this step is not required for Service Providers.

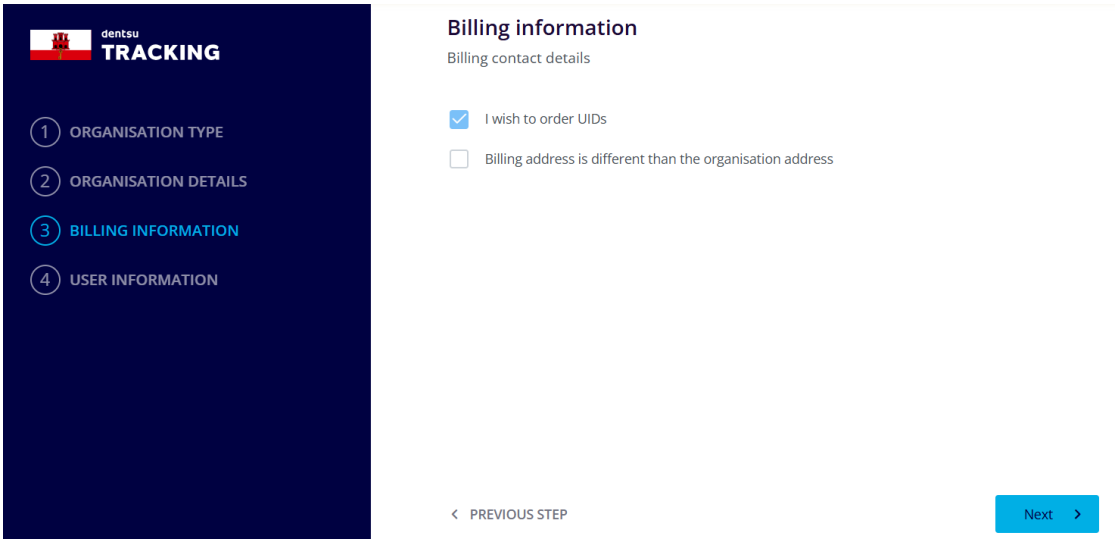

Figure 9 – Onboarding Wizard – Manufacturer/Importer view

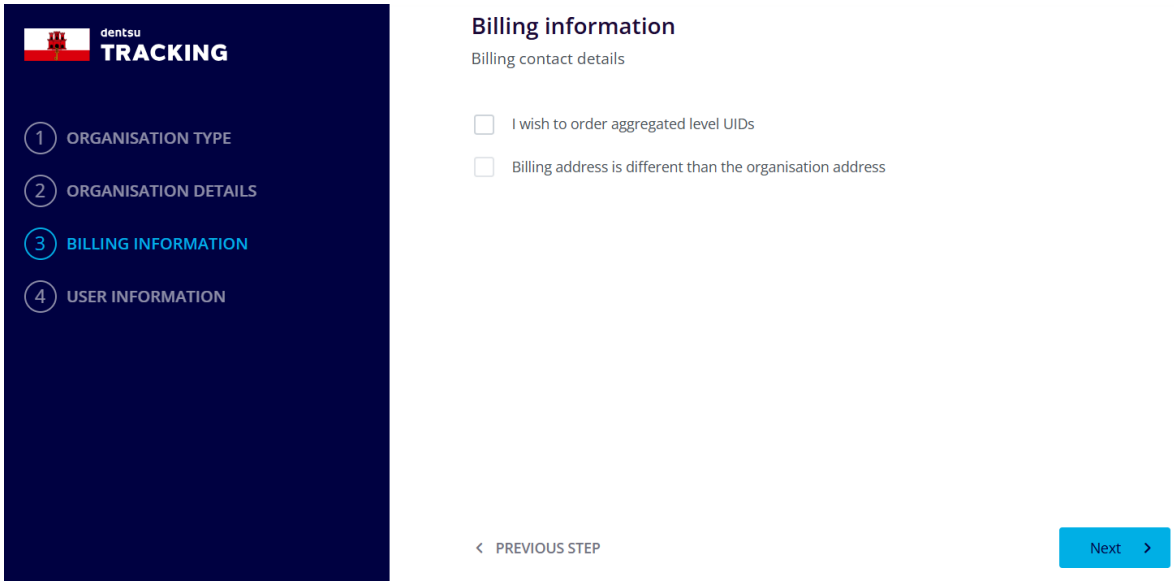

Figure 10 – Onboarding Wizard – Billing Information – Other Economic Operator view

"I wish to Order UIDs" or "I wish to Order aggregated level UIs":

- Select this option if the organisation wishes to be able to request / purchase UI codes. Dentsu will issue corresponding invoices for the submitted orders.

- Do not select this option if the organisation does not want to request UIs.

"Billing address is different than the Organisation Address":

This option only applies to organisation who wish to request / purchase upUI/aUI.

- Select this option if the billing address is different from the organisation's address provided during step 2. In this case an alternative address can be provided.

- Do not select this option if billing address and organisation address (provided in step 2) are identical.

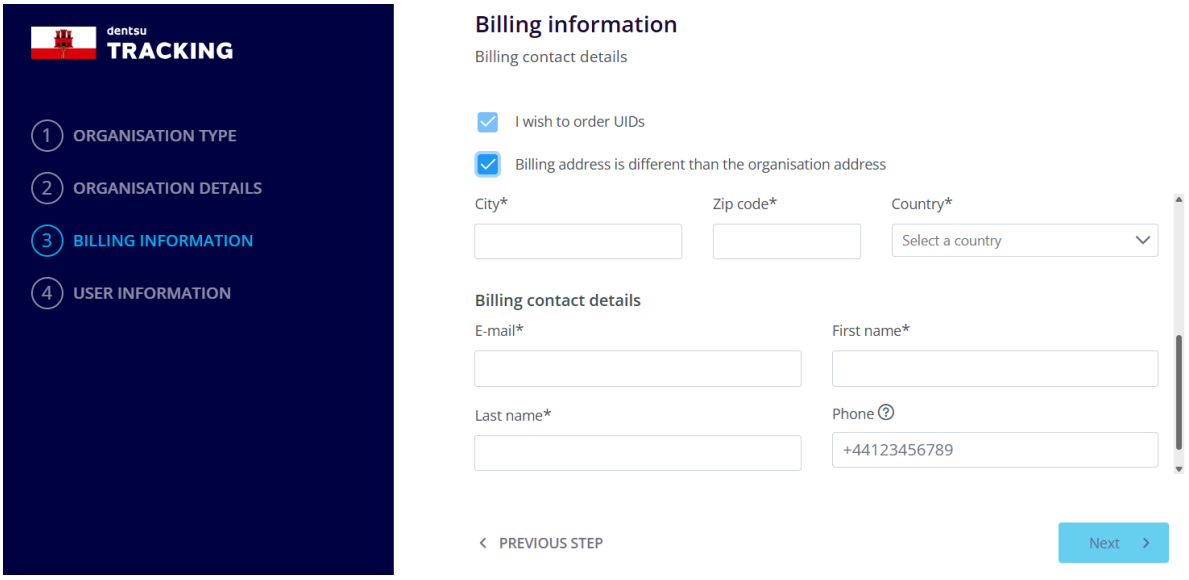

Figure 11 – Onboarding Wizard – Billing Information

Note that the following information can still be modified after the account has been approved:

- Billing Organisation name
- Relevant taxation number
- Street
- Number
- City
- Postal Code
- Country
- Billing Contact Details:
    - Email
    - First Name
    - Last Name
    - Phone

4) Step 4 – Complete all User Information details of the person managing the organisation account (note: additional users may be created after the account is approved):

- First Name
- Last Name

- Professional Email
- Confirm Professional Email
- Professional Phone Number
- Agree to our "Terms and Conditions"
- Accept Data Protection Policy (GDPR)

Note: It is mandatory to read the Terms and Conditions and to accept them. The terms and conditions frame the rights and obligations of both parties regarding the ID Issuer services. Accepting the privacy policy (GDPR) is also mandatory at the time of first connection.

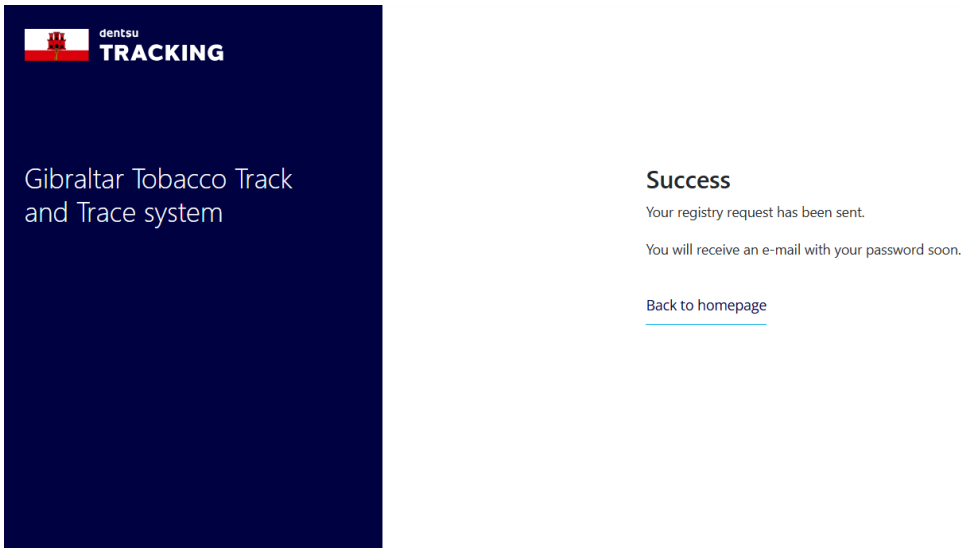

Figure 12 – Onboarding Wizard – Final

Once the registration process is completed, the user will get a "Welcome" email with an "auto-generated" password to be modified upon first login. At this point the user may already login into the system, but the account will be shown with status "processed".

Once Dentsu approves the Organisation, the user will get an email confirming that the account is active. As of that point, the user can perform operations in the GI ID Issuer Portal.

Every time that a new user is registered in the organization account, the new user will receive a welcome email with an "auto-generated" password.

Note: Creation of the organization account and user account does not trigger registration of an Economic Operator ID. Users must request Economic Operator IDs individually by completing the required information.

As of this stage, users can also request credentials to enable automatic integration with the Economic Operator's IT systems.

The user will also receive an additional e-mail from the Dentsu Tracking Support Portal (Service Now), confirming enrolment in the support portal where the user can open incident tickets.

4 Support process

For every user registered in the GI ID Issuer portal, a corresponding user gets created in the Service Now Portal of Dentsu Tracking.

Note: the passwords to access the GI ID Issuer and the Service Now support portal are different. For access related issues please contact the call centre or the generic support inbox at gi-tnt-support@dentsu.com.

Please check the Document Centre for announcements, and access to the Support Portal and GI ID Issuer Portal: https://www.gitobaccotracing.com on the "Service Portal" button at the top right corner.

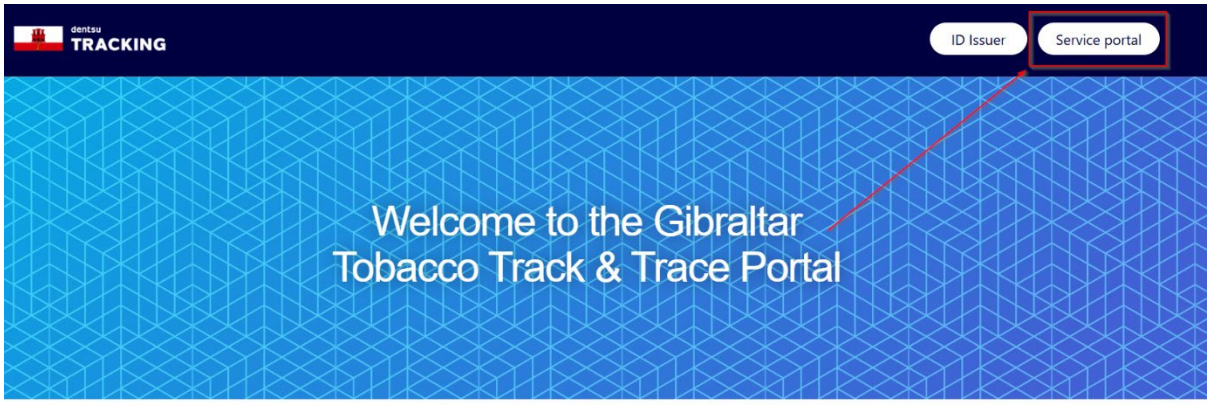

Figure 13 - GI Track & Trace Homepage - Access ID Issuer, Service Portal...

In any case, the Service Portal is accessible via the following link:

https://support.gitobaccotracing.com/giportal

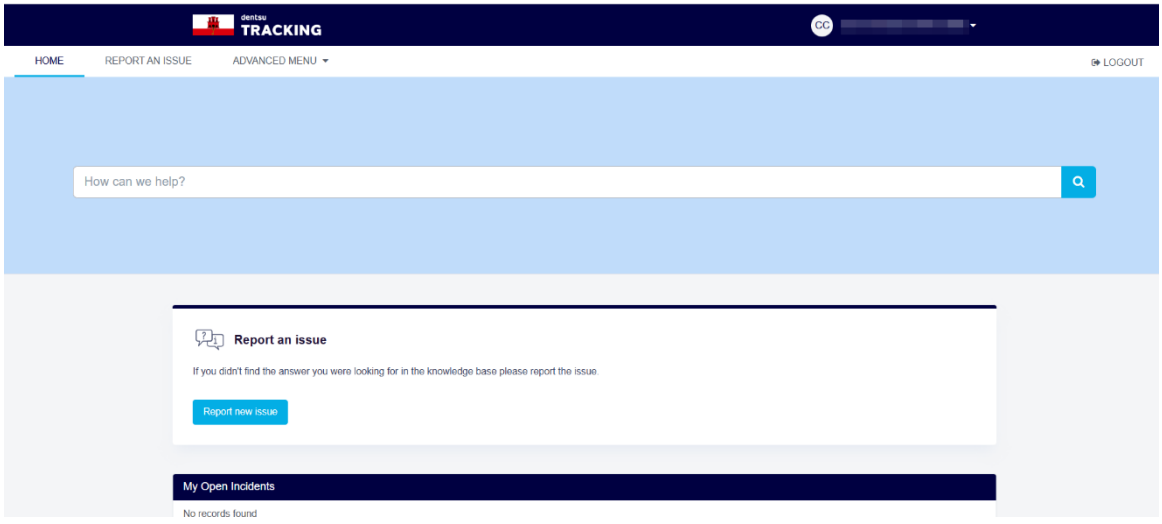

Figure 14 - GI Support Portal - After Login

Note: A separate user manual for the GI ID Issuer will be available in the "Technical documentation" section of the GI ID Issuer portal.

Dentsu provides business as usual support through Service Now and the Call Centre.

5 Code ordering modalities

5.1 Billing modalities for UI code orders

Dentsu Tracking charges a fee to economic operators for each requested UI code. Unit level and aggregated level UI codes (requested from the ID Issuer) are subject to the same fee. The fee was contractually agreed between Dentsu and HMGoG and may only be changed under very specific conditions and subject to approval by HMGoG.

Note: test UI codes used in the test environment (non-production environment) are not subject to any fee. Said test UI codes may not be used in production.

Invoices are issued to each organisation at the end of a month (the "Period of Reference"). The invoiced price for services will be based on the volume of new (unit level / aggregated level) Unique Identifiers requested over the preceding Period of Reference.

Invoices are sub-divided into Economic Operator IDs (linked to the respective order) and among others contain the relevant Period of Reference, the number of unit level Unique Identifiers concerned by that Period of Reference, the internal order number reference (if provided during the order request), and the amount due in EUR.

Note: all billing modalities and UI code ordering service terms are set out in the Terms & Conditions, which are currently accessible via the Web Interface and API environment.

Partial excerpt of the invoice format showing division by EO-ID:

Dentsu legal entity details		
		Client legal entity details
		Client purchase order number
Invoice number		Date
		Due date
		Payment term
Period of reference details		
Economic Operator 1	Order references	Volumes of UI
Economic Operator 2	Order references	Volumes of UI
Economic Operator 3	Order references	Volumes of UI
Economic Operator 4	Order references	Volumes of UI

Figure 15 – Partial Excerpt of the Invoice Format

At time of initial organisation creation in the system, the organisation will be subject to Dentsu's onboarding process for invoicing purposes. This process is handled via direct contact between the organisation and the Dentsu back-office team, running among others a standard due diligence exercise.

5.2 Terms and conditions

Dentsu issues Terms & Conditions that govern the overall use of the ID Issuer Services provided to Economic Operators, including applicable SLAs as laid down in CIR 2018/574 (as amended) and/or otherwise agreed with HMGoG. The Terms & Conditions must be accepted by all Economic Operators prior to using the production environment of the GI ID Issuer platform.

Economic Operators requesting UI codes from the ID Issuer must also accept additional terms governing the commercial aspects linked to the generation and issuing of UI codes, including invoicing and payment modalities. For those Economic Operators, the specific terms are included in the general Terms & Conditions document, which must be accepted by them prior to submitting any UI code order request. Acceptance takes place via ticking the box "Accept Dentsu Terms & Conditions" in the Web Interface or by declaring such acceptance in the API (via parameter: "Accept Terms") when submitting an order request.

The respective Terms & Conditions document is available to Economic Operators via the ID Issuer Web Interface.

5.3 SLA

5.3.1 Identifier Codes

Identifier codes (EOID, FID, MID) are issued instantly once an organisation account is available and the identifier code request was submitted and received correctly.

5.3.2 UI codes

CIR 2018/574 (as amended) sets out the applicable SLA for the UI code request, generation and issuing modalities.

Dentsu is required to generate and issue unit level and aggregated level UI codes based on the information submitted by economic operators as part of the request message (subject to applicable validation rules), whereas economic operators are responsible for the information contained in their request.

Dentsu will respond to correctly submitted order request within the legally defined timeframes:

- Electronic UIDs to be delivered within 2 Working Days.
- Electronic Aggregated UID to be delivered within 2 Working Days.

Dentsu starts the generating of UI codes 24 hours after receipt of the UI code ordering request, to allow Economic Operators to make use of their right to cancel an order within the legally permitted 24-hour time window.

However, Economic Operator may make use of a fast delivery option by selecting the field "No Cancellation" in the Web Interface or as a parameter in the API. Economic operators must note, however, that selecting this option automatically relieves them of the right to cancel an order within 24 hours. **UI codes requested with fast delivery option cannot be cancelled at any stage after submission of the order request.** There are no additional fees charged for the fast delivery option.

Note that there is no minimum number of codes to be requested in an order (the minimum is 1 code). There is a maximum of 1 million codes per order to prevent human error. If an Economic Operator needs more than 1 million codes at a time, please place multiple orders.

5.3.3 General availability of services

HMGoG contracted Dentsu to ensure that the ID Issuer services are subject to an uptime of 99.5%. Temporary unavailability due to planned maintenance agreed with HMGoG in advance, and duly communicated to economic operators does not qualify as downtime.

6 Format of UI codes

6.1 Issuing Agency Code / Company Number (ISO 15459-2)

According to applicable GI legislation, the GI ID Issuer must comply with ISO 15459-2. This ISO standard requires that the ID issuer is uniquely identifiable by means of an Issuing Agency Code / Company Number.

Dentsu uses the following ID Issuer prefix for the delivery of the GI ID Issuer:

QCGIT

6.2 Structure of Identifier codes

Economic Operator IDs, Facility IDs and Machine IDs must respect the following structure:

Figure 16 - Identifiers format

EOID: Economic Operator ID

FID: Facility ID

MID: Machine ID

Length: 13 Characters

The ID Issuer Identifier Code (5 characters) is identical for unit-level and aggregated-level UI codes. The serialised element is a randomised sequence of 8 characters to identify each identifier code uniquely in the system, in all capital letters to facilitate manual communication between parties.

Examples:

EOID: QCGITABC45THY

FID: QCGIT12ERBYNJ

MID: QCGIT89TTHVOK

6.3 Unit level Unique Identifiers

The UI code will follow the structure outlined in the next paragraph:

- **ID Issuer prefix**: 5 characters compliant with ISO/IEC 15459-2 (QCGIT).

- **Product code**: 3 characters, which compress the information provided during the request of the codes to meet regulatory requirements by means of an encrypted lookup.
- **Serial Number**: 8 to be able to randomize and make the code non predictable.

The UI code appears in three different versions:

- upUI(i): The format delivered to manufacturers and importers by the GI ID Issuer.
- upUI(L): The format of the code encoded by the manufacturer and importer into the data carrier.
- Human Readable (upUI(s)): The format printed next to the data carrier in plain text, where applicable and also delivered to manufacturers and importers by the GI ID Issuer.

Figure 17 - upUI(i) format

upUI(i): upUI identifier, the format delivered to the manufacturer or importer.

Sample: QCGIT1qW3e4RtOj5

Length: 16 Characters

Figure 18 - upUI(L) format

upUI(L): upUI long identifier, it's the upUI(i) plus the affixation of the Timestamp with format YYMMDDhh (Y – Year, M – Month, D – Day, h – hour)

Sample: QCGIT1qW3e4RtOj526010101

Length: 24 Characters

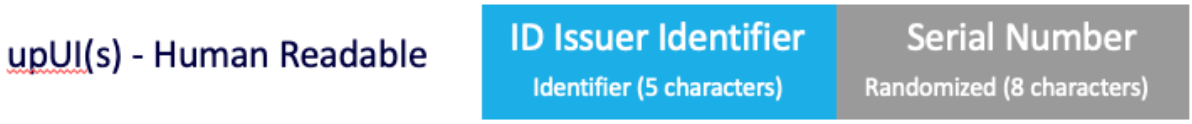

Figure 19 - Human Readable format – upUI(s)

Human Readable – upUI(s): The version to be printed so a human can read it. The human readable code must enable an electronic link to the corresponding UI code in the system.

Sample: QCGIT1qW3e4Rt

Length: 13 Characters

6.4 Aggregated level unique identifiers

The process to request aggregated level UI codes is almost identical to requesting unit level UI codes. Experience shows that most Economic Operators (manufacturers, importers, distributors) rely on the possibility to self-generate aggregated codes in accordance with internationally recognized ISO standards (e.g. ISO 15459 part 1 and part 4).

Where Economic Operators do not wish to self-generate, they can request aggregated level UI codes from the ID Issuer. The ID Issuer portal facilitates this option. Respective requests will be linked to the F-ID of the facility at which the aggregation event takes place.

The aggregated level UI code has the following structure:

- **ID Issuer prefix**: 5 characters based on ISO 15459-2 (QCGIT).
- **Aggregate Identification**: 1 to alleviate reading and identification (A).
- **Serial Number**: 8 characters to be able to randomize and make the code non predictable.

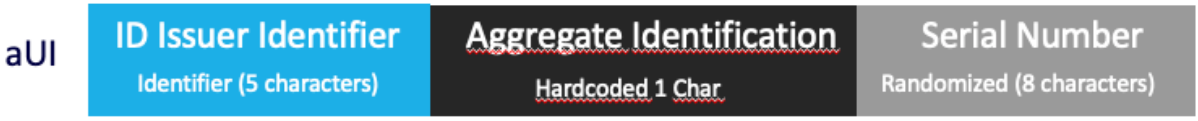

Figure 20 - Aggregated level UI format

aUI: Format delivered by the ID Issuer to the economic operator.

Sample: GCGITAK8UHTF9K

Length: 14 Characters

7 Clarification on Structure of unit-level unique identifiers

Data carriers must comply with the rules as set out by Article 21 [2] of CIR 2018/574 (as amended). The data carrier content must be preceded by a mandatory ISO data identifier 5R.

7.1 Clarification on Structure of unit-level unique identifiers (after encoding into a data carrier)

The purpose of this section is to clarify the use of data qualifiers as part of the UI, in line with CIR 2018/574 (as amended) and the applicable international ISO norms.

Please see the following table illustrating the structure of the UI (after encoding it into a data carrier), and the role of the ID issuer and economic operators in generating and/or applying different data elements and, where applicable, data qualifiers.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
Unique Identifier	Symbology Identifier	Mandatory Data Qualifier	ID Issuer Identification Code	Optional Data Qualifier	Serial Number	Optional Data Qualifier	Product Code	Optional Data Qualifier	Timestamp
Type	Qualifier	Qualifier	String (data Element)		String (data Element)		String (data Element)		String (data Element)
Position within the unique identifier	Fixed	Fixed	Fixed	Free	Free	Free	Free	Fixed	Fixed
Regulated by	Art. 21(1) and ID issuer's coding structure	Art.3(4), Art.8(1)(a), Art. 21(1) and ID issuer's coding structure	Art.3(4) and Art.8(1)(a)	Art. 21(1) and ID issuer's coding structure	Art.8(1)(b)	Art. 21(1) and ID issuer's coding structure	Art.8(1)(c)	Art. 21(1), Art. 21(4) and ID issuer's coding structure	Art.8(1)(d) and Art.21(4)
Applicable international standards	ISO/IEC 16022:2006, or ISO/IEC 18004:2015, or ISS DotCode Symbology Spec.	ISO 15459-2:2015 and ISO 15459-3:2014	ISO 15459-2:2015 and ISO 15459-3:2014						
Process	Applied bv EO	Applied bv EO	Generated by ID issuer	Applied bv EO	Generated by ID issuer	Applied bv EO	Generated by ID issuer	Applied bv EO	Applied bv EO
Transmission to repositories systems	No	No	Yes	No	Yes	No	Yes	No	Yes

Following Article 8(1)(a) -(c) of CIR 2018/574 (as amended), the following data elements (strings) should form part of the UI, as generated by the ID issuer:

- ID issuer identification code (subject to ISO 15459-2 and 3).
- Serial number.
- Product code.

In accordance with Articles 8(1)(d) and 21(4) of CIR 2018/574, manufacturers and importers must add the time stamp in the last position to the code generated by the ID issuer. The time stamp can be either encoded into the data carrier or be added separately from the data carrier in human-readable format. The time stamp format must be YYMMDDhh. Regardless of its format, the time stamp remains a part of the UI.

In accordance with Article 3(4) of CIR 2018/574, the ID issuer identification code should be assigned considering ISO/IEC 15459-2 and the latter should be read in conjunction with ISO/IEC 15459-3 laying down common rules on unique identification and data capture techniques. Similarly, the ID issuer identification code must always be preceded by a data qualifier, which must consist of digits and upper cases only.

The potential use of a data qualifier preceding the time stamp will also depend on whether an economic operator decides to rely on Article 21(4) of CIR 2018/574 (as amended). The application of said data qualifier should take place in accordance with the applicable coding structure published by the ID issuer in cooperation with its Issuing Agency.

To ensure positive validation by the GI Gateway, only the following data elements (strings), excluding the symbology identifier and any data qualifiers, are to be transmitted by economic operators as part of their reporting activity to the GI Track & Trace System:

- ID issuer identification code (without mandatory data qualifier).
- Serial number.
- Product code.
- Time stamp.

7.1.1 Encoding of unit level UIs (upUI)

upUIs must be encoded in the respective data carrier as follows:

Position 1: DI 5R

Position 2: fixed value QCGIT:

Position 3: the upUI (data element) as set out in 6.3, without preceding ID Issuer identifier 34htuskippp

Position 4: the time stamp in the format YYMMDDhh, which is optional acc. to derogation set out in Article 21(4)

Practical advice: replace upUI preceding ID Issuer identifier QCGIT by 5RQCGIT: and add the timestamp.

Example of the information to encode into the data carrier for an upUI

QCGIT34htuskippp with a production date of 1 July 2026, 6:00pm

included in the data carrier:

=> Data carrier content:

5RQCGIT:34htuskippp26070118

7.2 Clarification on Structure of aggregated-level unique identifiers (after encoding into a data carrier)

7.2.1 Encoding of aggregated level UIs (aUI) generated by the ID Issuer

For aggregated UIs, the rules on the use of data qualifiers apply by analogy, without timestamp.

This section applies only to aUIs generated by the ID Issuer and not to those aUIs that are self-generated by the Economic Operator

aUIs must be encoded in the data carrier as follows:

Position 1: DI 5R

Position 2: fixed value QCGIT:

Position 3: the aUI as set out in 6.4, without preceding ID Issuer identifier AK8UHTF9K

Position 5: optional additional Information as set out by Article 11(4) [2]

Practical advice: replace aUI preceding ID Issuer identifier QCGIT by 5RQCGIT: and add additional information, if desired.

Example of the information to encode into the data carrier for an aUI

QCGITAK8UHTF9K with an aggregation date of July 1st, 2022

7:00pm included in the data carrier:

=> Data carrier content:

5RQCGIT:AK8UHTF9K

7.2.2 Self-generated aggregated UIs

Self-generated aUIs must only provide for unique identification of the traceable item. Therefore, any additional information added to the aggregated level UI, as provided in Article 11(4) of CIR 2018/574 (as amended), must not be transmitted by economic operators as part of their reporting activity to the GI Track & Trace System.

Example 1: GS1 DataMatrix encoding Global Trade Item Number with Serial Number (SGTIN)

Aggregate Unique Identifier for standard trade item grouping using GS1 Application Identifiers (01) for GTIN and (21) for Serial Number.

	(1)	(2)	(3)	(4)	(5)
Unique Identifier	Symbology Identifier	Data Qualifier	GTIN	Data Qualifier	Serial Number
Position within the unique identifier	**Fixed**	**Fixed**	**Fixed**	**Fixed**	**Fixed**
Applicable international standards:	ISO/IEC 16022:2006, ISO/IEC 18004:2015, or ISO/IEC 15417:2007	ISO 15459-2:2015, ISO 15459-3:2014, ISO/IEC 15459-4: 2014 Section 4.1.2 (normative), ISO/IEC 15459-6:2014 Section 5 (normative) and Annex B (informative), and the GS1 General Specifications V.19 (or latest equivalents)			
Values	]d2	01	01234567891231	21	456FGRD66
Process	☐ Applied by EO ☐ Symbology Identifiers are transmitted by scanners based on 'start character patterns' that must be followed when printing the barcode. See barcode specifications for specific patterns required to signal GS1 formatted data.				
Transmission to repositories systems	No	No	Yes	No	Yes
aUI			01234567891231		456FGRD66

Example 2: GS1 DataMatrix encoding SGTIN (required for aUI) with additional information permitted, but not required.

Aggregate level Unique Identifier for standard trade item grouping adding GS1 Application Identifier (240) Additional Product ID assigned by the manufacturer to Example 1.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Unique Identifier	Symbology Identifier	Data Qualifier	GTIN	Data Qualifier	Serial Number	Character required by GS1 after variable	Data Qualifier for optional attribute added by	Additional product identification assigned by the

						length fields	the manufacturer	manufacturer
Position within the unique identifier	**Fixed**	**Fixed**	**Fixed**	**Fixed**	**Fixed**	Per rules within GS1 General Specifications V19 or latest equivalent		
Applicable international standards:	ISO/IEC 16022:2006, 18004:2015, or 15417:2007	ISO 15459-2:2015, ISO 15459-3:2014, ISO/IEC 15459-4: 2014 Section 4.1.2 (normative), ISO/IEC 15459-6:2014 Section 5 (normative) and Annex B (informative), and the GS1 General Specifications V.19 (or latest equivalents)						
Values	]d2	01	01234567891231	21	456FGRD66	FNC1 or <GS>	240	ED1234
Process	☐ Applied by EO ☐							
Transmission to repositories systems	No	No	Yes	No	Yes	No	No	No
aUI			01234567891231		456FGRD66			

Example 3: GS1-128 encoding Serial Shipping Container Code (SSCC)

Aggregate level Unique Identifier for transport (logistic) units using GS1 Application Identifier (00) SSCC.

	(1)	(2)	(3)
Unique Identifier	Symbology Identifier	Data Qualifier	SSCC
Position within the unique identifier	**Fixed**	**Fixed**	**Fixed**
Applicable international standards:	ISO/IEC 16022:2006, ISO/IEC 18004:2015, or ISO/IEC 15417:2007	ISO 15459-2:2015, ISO 15459-3:2014, ISO/IEC 15459-1: 2014 Section 4 (normative) and Annexes A and B (informative), and the GS1 General Specifications V.19 (or latest equivalents)	
Values	]C1	00	123456789123456789
Process	☐ Applied by EO ☐		
Transmission to repositories systems	No	No	Yes
aUI			123456789123456789

Example 4: Code 128 bar code symbol with the qualifier of ASC MH10 Data Identifier "J".

As defined in ISO/IEC 15459-1:2014 Annex A (informative) Unique identification for transport units section A.3 ASC MH10 unique identification for transport units.

	(1)	(2)	(3)
Unique Identifier	Symbology Identifier	ASC MH 10 Data Identifiers	
Values	]C0	J	JNLY1234567890
Process	Applied bv EO	Applied bv EO	Applied bv EO
Transmission to repositories systems	No	No	Yes
aUI			JNLY1234567890

7.3 General Decoding Rules

The GI Gateway expects the information without Data Qualifiers / Application Identifiers contained in the data carrier. Any party processing information intended for transmission to the GI Gateway contained in a data carrier, must strip the preceding control information beforehand.

7.3.1 Unit pack level Unique Identifiers (upUI)

Decoding example for data carrier content:

5RQCGIT:34htuskippp26070118

=> Information for transmission to repository system:

QCGIT34htuskippp26070118

7.3.2 Aggregated level Unique Identifiers (aUI)

Decoding example for data carrier content:

5RQCGIT:AK8UHTF9K

=> Information for transmission to repository system:

QCGITAK8UHTF9K

8 GI ID Issuer Interface

8.1 Web Portal

A Web Portal is available to all Economic Operators where they can perform operations via a regular web application.

A separate User Manual for Economic Operators is available at the web portal via the "Technical Documentation" section.

The following operations can be performed via the web portal:

- Manage users for the organisation (Create additional users to access the GI ID Issuer
- Request / manage API Credentials (OAUTH2 based, client + secret)
- Create, modify or de-register Economic Operator IDs (registered for the company or on-behalf)
- Create, modify or de-register Facility IDs (registered for the company or on-behalf)
- Create, modify or de-register Machine IDs.
- Verify Economic Operator IDs, Facility IDs and Machine IDs (across the entire registry, verifying their existence and validity)
- Dashboard providing a summary of the code ordering activity
- Request upUIs electronically
- Request aUIs electronically

8.2 API

This section details the GI ID Issuer API. A swagger definition of the API is available:

For Pre-Production: https://api.idissuer.pre.gitobaccotracing.com/swagger/index.html

For Production: https://api.idissuer.prod.gitobaccotracing.com/swagger/index.html

8.2.1 How to request credentials

Users (Economic Operator or Service Provider) have access to the web interface after registering in the GI ID Issuer system. The web interface contains a dedicated section to request API credentials.

If any issue is encountered during this process, please contact the support team via the support portal: https://support.gitobaccotracing.com/giportal

8.2.2 EO_CODE

An EO_CODE will be generated for each Economic Operator. This code must always be transmitted as part of the message requesting changes to the identifier codes. The EO_CODE is unique per Organisation/Account.

Field	Data Type	Description	Implementation
EO_CODE	Text(10)	Economic operator's confirmation code provided in	10-character alphanumeric random.

		response to the registration of economic operator.	The EO_CODE will be unique per Organisation (account). When creating an EOID the EO_CODE of the organisation will be returned.

8.2.3 Country Codes

List of permitted two digits country codes for the INTENDED_MARKET fields and other fields using the Country type.

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	Åland 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
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
XI	Northern Ireland (in accordance with the Northern Ireland Protocol)
XK	Kosovo
XZ	International waters

YT	Mayotte
ZA	South Africa

ZM	Zambia
ZW	Zimbabwe

9 Message level validations

The validations described in this section affect all messages in the GI ID Issuer API.

KEY	**Error Code**	**Error Description**	**http Status**
VAL_FIE_FORMAT	INVALID_INPUT_FORMAT REQUIRED_FIELD_FAILED_VALIDATION MAX_LENGTH_FAILED_VALIDATION MIN_LENGTH_FAILED_VALIDATION	The field {xx} does not meet the required format	400
VAL_SEC_TOKEN	INVALID_OR_EXPIRED_TOKEN	The token is invalid or has been expired	401
VAL_MSG_CODE	INVALID_EO_CODE	The EO code is wrong or does not exist	400
VAL_FIE_REF	FAILED_VALIDATION	The field {xx} is not related to field {xx}	400
VAL_ENT_EXIST_EOID	EOID_NOT_EXIST_OR_ACTIVE (applies to all fields except the fields Other_EOID_N)	The EO {xx} does not exist or is not active	400
VAL_ENT_EXIST_FID	FID_NOT_EXIST_OR_ACTIVE (applies to all fields except the fields Other_FID_N)	The FID {xx} does not exist or is not active	400
VAL_ENT_EXIST_MID	MID_NOT_EXIST_OR_ACTIVE	The MID {xx} does not exist or is not active	400
VAL_ENT_ACTIVE_EOID	EOID_NOT_EXIST_OR_ACTIVE (applies to all fields except the fields Other_EOID_N)	The EO {xx} does not exist or is not active	400
VAL_ENT_ACTIVE_FID	FID_NOT_EXIST_OR_ACTIVE (applies to all fields except the fields Other_FID_N)	The FID {xx} does not exist or is not active	400

VAL_ENT_ACTIVE_MID	MID_NOT_EXIST_OR_ACTIVE	The MID {xx} does not exist or is not active	400
VAL_NOT_FOUND	ENTITY_NOT_FOUND (applies to get, delete, and put methods where there is no data to be returned)	No data available	404
VAL_ENT_EXIST_ORG	ORG_NOT_EXIST	The organisation {xx} is not registered in the system.	404

General errors

KEY	**Error Code**	**Error Description**	**http Status**
SYSTEM_ERROR	SYSTEM_ERROR	There was an error processing the request {internalID}	500

Note: The responses (acknowledgment message), if negative, contain a list of all errors identified.

10 REGISTRY OPERATIONS

10.1 GI ID Issuer API - Register Economic Operator (REO)

Use this method to register a new Economic Operator ID.

POST/economic-operators Create economic operator

10.1.1 Request Definition – "EORegistryRequest"

KEY	**Base Type**	**Type in TPD**	**Priority in TPD**	**Values in TPD**
EO_Address	string	Text(300)	M	Maps to "EO_street" in the EU System in combination with EO_Address_StreetOne
EO_CountryReg	string	Text(2)	M	Country list is defined in GI Gateway Technical Specifications document
EO_Email	string	Text(5000)	M	Regex protected:

				Validation RegEx: ^((([a-zA-Z]|\d|[!#\$%&'*\+\-\/=\?\^_`{|}~]|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])+(\.([a-zA-Z]|\d|[!#\$%&'\*\+\-\/=\?\^_`{|}~]|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])+)*)|((\x22)(((\x20|\x09)*(\x0d\x0a))?(\x20|\x09)+)?(([\x01-\x08\x0b\x0c\x0e-\x1f\x7f]|\x21|[\x23-\x5b]|[\x5d-\x7e]|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])|(\\([\x01-\x09\x0b\x0c\x0d-\x7f]|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])))*(((\x20|\x09)*(\x0d\x0a))?(\x20|\x09)+)?(\x22)))@((([a-zA-Z]|\d|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])|(([a-zA-Z]|\d|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])([a-zA-Z]|\d|-|\.|_|~|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])*([a-zA-Z]|\d|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF]))\.)+(([a-zA-Z]|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])|(([a-zA-Z]|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])([a-zA-Z]|\d|-

				|\.|_|~|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])*([a-zA-Z]|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF]))\.?$
EO_ExciseNumber1	boolean	boolean	M	0 - No SEED number 1 - SEED number exists
EO_ExciseNumber2	string	SEED	M, if EO_ExciseNumber1 = 1	Excise number composed of: (a) country name coded with ISO-3166-1:2013 alpha-2 (or its latest equivalent) (e.g. 'LU') and (b) eleven alpha numeric characters, if needed, padded to the left with zeroes (e.g. '00000987ABC'). Sample: 'LU00000987ABC'
EO_Name1	string	text(100)	M	
EO_Name2	string	text(100)	O	
OtherEOID_N	array strings	array of EOIDs	M, if OtherEOID_R = 1	
OtherEOID_R	boolean	boolean	M	0– No | 1– Yes
Reg_3RD	boolean	boolean	M	0– No | 1– Yes
Reg_EOID	string	EOID	M, if Reg_3RD = 1	
TAX_N	string	Text(20)	M, if VAT_R = 0	
VAT_N	string	Text(20)	M, if VAT_R = 1	
VAT_R	boolean	boolean	M	0 - No VAT Registration 1 - VAT number exists
EO_Address_Name	string	Text(5000)	O	
EO_Address_City	string	Text(5000)	M	

EO_Address_StreetOne	string	Text(5000)	M	
EO_Address_StreetTwo	string	Text(5000)	O	
EO_Address_PostCode	string	Text(5000)	O	

10.1.2 Response Definition – "EORegistryResponse"

KEY	**Base Type**
EO_CODE	EO_CODE
EO_ID	EOID

Status Code	**Description**
201	EORegistryResponse (Economic operator is successfully created)
400	List[{ Error_Code: string, Error_Description: string}]
401	VAL_SEC_TOKEN
403	VAL_SEC_CLAIM
404	VAL_ENT_EXIST_ORG
500	SYSTEM_ERROR

10.1.3 Specific Message Validations

KEY	**Error Code**	**Error Description**	**http Status**
VAL_SEC_CLAIM	CLAIM_VALIDATION_FAILED	The client has no permission to create an EO	403
VAL_EXIST_OTHER_EOID	OTHER_EOID_ALREADY_EXIST	One or more of the EOIDs in the Other_EOID_N field already exist: {XX}	409

10.2 GI ID Issuer API - Get Economic Operator (GEO)

Use this method to retrieve information related to an Economic Operator ID of your Organisation.

GET/economic-operators/{EO_ID} Get economic operator

10.2.1 Request Definition – "GetEORequest"

Name	Description
EO_ID* (path)	Economic Operator ID (EOID)

10.2.2 Response Definition - "EconomicOperatorView"

KEY	Base Type	Priority in TPD
EO_ID	EOID	M
EO_Address	string	M
EO_CountryReg	string	M
EO_Email	string	M
EO_ExciseNumber1	boolean	M
EO_ExciseNumber2	string	M, if EO_ExciseNumber1 = 1
EO_Name1	string	M
EO_Name2	string	O
OtherEOID_N	array strings	M, if OtherEOID_R = 1
OtherEOID_R	boolean	M
Reg_3RD	boolean	M
Reg_EOID	string	M, if Reg_3RD = 1
TAX_N	string	M, if VAT_R = 0
VAT_N	string	M, if VAT_R = 1
VAT_R	boolean	M
EO_Address_Name	string	O

EO_Address_City	string	M
EO_Address_StreetOne	string	M
EO_Address_StreetTwo	Text(5000)	O
EO_Address_PostCode	Text(5000)	O
EO_Active_Status	boolean	M

Status Code	**Description**
200	returns economic operator: "EconomicOperatorView"
400	List[{ Error_Code: string, Error_Description: string}]
401	VAL_SEC_TOKEN
403	VAL_SEC_CLAIM
500	SYSTEM_ERROR

10.2.3 Specific Message Validations

KEY	**Error Code**	**Error Description**	**http Status**
VAL_SEC_CLAIM	CLAIM_VALIDATION_FAILED	The client has no permission to get EO information	403

10.3 GI ID Issuer API - List Economic Operators (LEO)

Use this method to retrieve information related to all Economic Operator IDs that meet the query parameters for your Organisation.

GET/economic-operators Get economic operators list

10.3.1 Query Parameters

Name	**Type**	**Priority**	**Description**

EO_Active_Status	boolean	O	Select only activated/deactivated EO
EO_CountryReg	array[string]	O	Select only EO which are registered at the requested country/ies
Page	string	O	Page number, default 1
Reg_EOID	string	O	Select EO by Reg_EOID field
Sort	string	O	Sort the result by the selected option (Available values: ID, NAME_ASC, NAME_DESC (not include in the query options), COUNTRY_ASC, COUNTRY_DESC, STATUS) Note: NAME_ASC and NAME_DESC should be based on the field EO_Name1. COUNTRY_ASC and COUNTRY_DESC should be based on the field EO_CountryReg

10.3.2 Response Definition – ListOfEconomicOperatorView

EO_List*	List<EconomicOperatorView>
Current_Page	int
Total_Pages	int

Status Code	**Description**
200	ListOfEconomicOperatorView
400	List[{Error_Code: string, Error_Description: string}]
401	VAL_SEC_TOKEN
403	VAL_SEC_CLAIM
500	SYSTEM_ERROR

10.3.3 Specific Message Validations

KEY	Error Code	Error Description	http Status
VAL_SEC_CLAIM	CLAIM_VALIDATION_FAILED	The client has no permission to get EO information	403

10.4 GI ID Issuer API - Correct Economic Operator (CEO)

Use this method to modify the information related to an Economic Operator ID of your Organisation.

PUT/economic-operators/{EO_ID} Edit economic operator

10.4.1 Request Definition - "EconomicOperatorEditRequest"

KEY	Base Type	Type in TPD	Priority in TPD	Values in TPD
EO_ID	String (path)	EOID	M (not editable)	
EO_CODE	string	string	M (not editable)	
EO_Address	string	Text(300)	M	
EO_CountryReg	string	Text(2)	M	Country list is defined in GI Gateway Technical Specifications document
EO_Email	string	Text(5000)	M	
EO_ExciseNumber1	boolean	boolean	M	0 - No SEED number 1 - SEED number exists
EO_ExciseNumber2	string	SEED	M, if EO_ExciseNumber1 = 1	Excise number composed of: (a) country name coded with ISO-3166- 1:2013 alpha-2 (or its latest equivalent)

				(e.g. 'LU') and (b) eleven alpha numeric characters, if needed, padded to the left with zeroes (e.g. '00000987ABC'). Sample: 'LU00000987ABC'
EO_Name1	string	Text(100)	M	
EO_Name2	string	Text (100)	O	
OtherEOID_N	array strings	array of EOIDs	M, if OtherEOID_R = 1	
OtherEOID_R	boolean	boolean	M	0– No | 1– Yes
Reg_3RD	boolean	boolean	M	0– No | 1– Yes
Reg_EOID	string	EOID	M, if Reg_3RD = 1	
TAX_N	string	Text(20)	M, if VAT_R = 0	
VAT_N	string	Text(20)	M, if VAT_R = 1	
VAT_R	boolean	boolean	M	0 - No VAT Registration 1 - VAT number exists
EO_Address_Name	string	Text(5000)	O	
EO_Address_City	string	Text(5000)	M	
EO_Address_StreetOne	string	Text(5000)	M	
EO_Address_StreetTwo	Text(5000)	Text(5000)	O	
EO_Address_PostCode	Text(5000)	Text(5000)	O	

10.4.2 Response Definition – "EconomicOperatorEditResponse"

Status Code	**Description**
204	Economic operator modified successfully
400	List[{ Error_Code: string, Error_Description: string}]

401	VAL_SEC_TOKEN
403	VAL_SEC_CLAIM
500	SYSTEM_ERROR

10.4.3 Specific Message Validations

KEY	Error Code	Error Description	http Status
VAL_SEC_CLAIM	CLAIM_VALIDATION_FAILED	The client has no permission to modify an EO	403
VAL_NOT_EDIT	FAILED_VALIDATION	The field {XX} is not editable	400

10.5 GI ID Issuer API - De-Registration of Economic Operator (DEO)

Use this method to de-register an Economic Operator ID of your Organisation.

DELETE/economic-operators/{EO_ID} De-Register an Economic Operator and related entities

When de-registering an Economic Operator, all facilities attached to it will be de-registered automatically.

Machines and machine parts associated to the facilities associated with the Economic Operator will be de-registered with the only exception of machine parts of type “Mobile”.

10.5.1 Request Definition - "EconomicOperatorDeregistrationRequest"

KEY	Base Type	Type in TPD	Priority in TPD	Values in TPD
EO_ID (path)	EOID	EOID	M	
EO_CODE	string	EO_CODE	M	
Reg_3RD	boolean	Boolean	M	0 - No 1 - Yes

Reg_EOID	string	EOID	M	M, if Reg_3RD = 1

10.5.2 Response Definition – "EconomicOperatorDeregistrationResponse"

Parameter	**Base Type**	**TPD Type**	**TPD Priority**	**Value**
EO_ID	string	EOID	M	
EO_Active_Status	boolean	boolean	M	Active / Inactive
Inactive_FIDs	array	FID	M	FIDs children of this EO that have been deactivated
Inactive_MIDs	array	MID	M	MIDs children of each FIDs children of this EO that have been deactivated

Status Code	**Description**
200	EconomicOperatorDeregistrationResponse
400	List[{ Error_Code: string, Error_Description: string}]
401	VAL_SEC_TOKEN
403	VAL_SEC_CLAIM
500	SYSTEM_ERROR

10.5.3 Specific Message Validations

KEY	**Error Code**	**Error Description**	**http Status**
VAL_SEC_CLAIM	CLAIM_VALIDATION_FAILED	The client has no permission to deactivate an EO	403

10.6 GI ID Issuer API - Register Facility (RFA)

Use this method to register a Facility ID for your Organisation.

POST/facilities Create a facility

10.6.1 Request Definition - "FacilityRequest"

Dentsu (Name - Type)	**Base Type**	**TPD Type**	**TPD Priority (O = optional, M = Mandatory)**	**Values**
EO_CODE	string	EO_CODE	M	
EO_ID	string	EOID	M	
F_Address	string	Text(5000)	M	
F_Address_City	string	Text(5000)	M	
F_Address_Name	string	Text(5000)	O	
F_Address_PostCode	string	Text(5000)	O	
F_Address_StreetOne	string	Text(5000)	M	
F_Address_StreetTwo	string	Text(5000)	O	
F_Country	string	Country	M	
F_ExciseNumber1	boolean	Boolean	M	0 - No SEED number 1 - SEED number exists
F_ExciseNumber2	string	SEED	M if F_ExciseNumber1 = true	
F_Name	string	Text(500)	O	
F_Status	boolean	Boolean	M	0 - No 1 - Yes
F_Type	int	Integer	M	1 - Manufacturing site with warehouse 2 - Standalone warehouse 3 - First retail outlet 4 - Other

F_Type_Other	string	Text(5000)	M, if F_Type = 4	
OtherFID_N	Array	FID	M, if OtherFID_R = 1	
OtherFID_R	boolean	Boolean	M	0 - No 1 - Yes
Reg_3RD	boolean	Boolean	M	0 - No 1 - Yes (possible only if F_Type = 3)
Reg_EOID	string	EOID	M, if Reg_3RD = 1	

10.6.2 Response Definition – "FacilityRegistryResponse"

F_ID	FID

Status Code	**Description**
201	FacilityRegistryResponse
400	List[{ Error_Code: string, Error_Description: string}]
401	VAL_SEC_TOKEN
403	VAL_SEC_CLAIM
404	VAL_ENT_EXIST_ORG
500	SYSTEM_ERROR

10.6.3 Specific Message Validations

KEY	**Error Code**	**Error Description**	**http Status**
VAL_SEC_CLAIM	CLAIM_VALIDATION_FAILED	The client has no permission to create a facility	403
VAL_EXIST_OTHER_FID	OTHER_FID_ALREADY_EXIST	One or more of the FIDs in the Other_FID_N field already exist: {XX}	409

10.7 GI ID Issuer API - Get Facility (GFA)

Use this method to request information related to a Facility ID of your Organisation.

GET/facilities/{F_ID} Get facility

10.7.1 Request Definition – "GetFacilityRequest"

Name	Description
F_ID* (path)	FID

10.7.2 Response Definition - "FacilityView"

Dentsu (Name - Type)	Base Type	TPD Priority (O = optional, M = Mandatory)
EO_ID	EOID	M
F_ID	FID	M
EO_Name1	string	M
F_Address	string	M
F_Address_City	string	M
F_Address_Name	string	O
F_Address_PostCode	string	O
F_Address_StreetOne	string	M
F_Address_StreetTwo	string	O
F_Country	string	M
F_ExciseNumber1	boolean	M
F_ExciseNumber2	string	M if F_ExciseNumber1 = true
F_Name	string	O
F_Status	boolean	M
F_Active_Status	boolean	M

F_Type	int	M
F_Type_Other	string	M, if F_Type = 4
OtherFID_N	Array	M, if OtherFID_R = 1
OtherFID_R	boolean	M
Reg_3RD	boolean	M
Reg_EOID	string	M, if Reg_3RD = 1

Responses:

Status Code	**Description**
200	FacilityView
400	List[{ Error_Code: string, Error_Description: string}]
401	VAL_SEC_TOKEN
403	VAL_SEC_CLAIM
500	SYSTEM_ERROR

10.7.3 Specific Message Validations

KEY	**Error Code**	**Error Description**	**http Status**
VAL_SEC_CLAIM	CLAIM_VALIDATION_FAILED	The client has no permission to get facility information	403

10.8 GI ID Issuer API - List Facilities (LFA)

Use this method to request information related to all Facility IDs of your Organisation that meet the query parameters.

GET/facilities Get facilities list

10.8.1 Query parameters

	Name	**Type**	**Description**

query	F_Active_Status	boolean	Select only activated/deactivated facilities
query	F_Country	array[string]	Select only facilities registered in given countries
query	EO_ID	EOID	Select only facilities linked to given EOID
query	Page	string	Page number, default 1
query	F_Type	array[integer]	1 - Manufacturing site with warehouse 2 - Standalone warehouse 3 - First retail outlet 4 - Other
query	Sort		EO_ID, F_TYPE, COUNTRY_ASC, COUNTRY_DESC, STATUS

10.8.2 Response Definition – ListOfFacilityView

Facility_List*	List<FacilityView>
Current_Page	int
Total_Pages	int

Responses

Status Code	**Description**
200	EntityListOfFacilityView
400	List[{ Error_Code: string, Error_Description: string}]
401	VAL_SEC_TOKEN
403	VAL_SEC_CLAIM
500	SYSTEM_ERROR

10.8.3 Specific Message Validations

KEY	**Error Code**	**Error Description**	**http Status**

VAL_SEC_CLAIM	CLAIM_VALIDATION_FAILED	The client has no permission to get facility information	403

10.9 GI ID Issuer API - Correct Facility (CFA)

Use this method to correct information related to a Facility ID of your Organisation.

PUT/facilities/{F_ID} Edit facility

10.9.1 Request Definition – "FacilityEditRequest"

Parameter	**Base Type**	**TPD Type**	**TPD Priority**	**Value**
EO_CODE	string	EO_CODE	M (not editable)	
EO_ID	string	EOID	M (not editable)	
F_ID (path)	string	FID	M (not editable)	
F_Name	string	Text(500)	O	
F_Address	string	Text(5000)	M	
F_Address_Name	string	Text(5000)	O	
F_Address_StreetOne	string	Text(5000)	M	
F_Address_StreetTwo	string	Text(5000)	O	
F_Address_City	string	Text(5000)	M	
F_Address_PostCode	string	Text(5000)	O	
F_Country	string	Country	M	
F_ExciseNumber1	boolean	boolean	M	0 - No SEED Number 1 - SEED number exists
F_ExciseNumber2-string	string	SEED	M, if F_ExciseNumber1 = 1	
F_Status	boolean	Boolean	M	0 - No 1 - Yes

F_Type	int	Integer	M	
F_Type_Other	string	Text(5000)	M, if F_Type = 4	
OtherFID_N	Array	FID	M, if OtherFID_R = 1	
OtherFID_R	boolean	Boolean	M	0 - No 1 - Yes
Reg_3RD	boolean	Boolean	M	0 - No 1 - Yes (possible only if F_Type = 3)
Reg_EOID	string	EOID	M, if Reg_3RD = 1	

10.9.2 Response Definition – "FacilityEditResponse"

Status Code	**Description**
204	Facility modified successfully
400	List[{ Error_Code: string, Error_Description: string}]
401	VAL_SEC_TOKEN
403	VAL_SEC_CLAIM
500	SYSTEM_ERROR

10.9.3 Specific Message Validations

KEY	**Error Code**	**Error Description**	**http Status**
VAL_SEC_CLAIM	CLAIM_VALIDATION_FAILED	The client has no permission to modify a Facility	403
VAL_NOT_EDIT	FAILED_VALIDATION	The field {XX} is not editable	400

10.10 GI ID Issuer API - De-registration of Facility (DFA)

Use this method to de-register a Facility ID of your Organisation.

DELETE/facilities/{F_ID} De-register facility

When de-registering a Facility, machines and machine parts associated to the facility will be de-registered with the only exception of machine parts of type “Mobile”.

10.10.1 Request Definition - "FacilityDeregistrationRequest"

KEY	Base Type	Type in TPD	Priority in TPD	Values in TPD
EO_CODE	string	EO_CODE	M	
EO_ID	string	EOID	M	
Reg_3RD	boolean	boolean	M	0 - No 1 - Yes
Reg_EOID	string	EOID	M	M, if Reg_3RD = 1
F_ID (path)	string	FID	M	

10.10.2 Response Definition – “FacilityDeregistrationResponse”

Parameter	Base Type	TPD Type	TPD Priority	Value
F_ID	string	FID	M	
F_Active_Status	boolean	boolean	M	Active / Inactive
Inactive_MIDs	array	MID	M	MID children of this FID that have been deactivated

Status Code	Description
200	FacilityDeactivationResponse
400	List[{ Error_Code: string, Error_Description: string}]
401	VAL_SEC_TOKEN

403	VAL_SEC_CLAIM
500	SYSTEM_ERROR

10.10.3 Specific Message Validations

KEY	Error Code	Error Description	http Status
VAL_SEC_CLAIM	CLAIM_VALIDATION_FAILED	The user has no permission to deactivate a facility	403

10.11 GI ID Issuer API - Register Machine (RMA)

Use this method to register a Machine ID for your Organisation

POST/machines

10.11.1 Request Definition - "MachineRegistryRequest"

Dentsu (Name - Type)	Base Type	TPD Type	TPD Priority (O = optional, M = Mandatory)	Values
EO_CODE	string	EO_CODE	M	
EO_ID	string	EOID	M	
F_ID	string	FID	M	
M_Producer	string	Text(200)	M, if M_entirety = 1	
M_Model	string	Text(200)	M, if M_entirety = 1	
M_Number	string	Text(200)	M, if M_entirety = 1	
M_Capacity	integer	integer	M, if M_entirety = 1	
M_Name	string	Text(500)	M	

M_ATD	string	Text(100)	M, if M_entirety = 1 and M_parts = 0	
PrevMID_B	boolean	boolean	M	0 – No (first time registration) 1 – Yes
PrevMID_ID	MID	MID	M, if PrevMID_B = 1	
M_entirety	boolean	boolean	M	0 – No (machine part) 1 – Yes (machine)
P_Producer	String	Text(200)	M, if M_entirety = 0	
P_Model	String	Text(200)	M, if M_entirety = 0	
P_Number	String	Text(200)	M, if M_entirety = 0	
P_Mobile	boolean	Boolean	M, if M_entirety = 0	0 – No (fixed part) 1 – Yes (mobile part)
P_ATD1	Boolean	Boolean	M, if M_entirety = 0	
P_ATD2	string	Text(100)	M, if M_entirety = 0 and P_ATD1 = 1	
P_Description	string	Text(500)	O	
M_parts	boolean	boolean	M, if M_entirety = 1	0 – No (not multiple separately parts) 1 – Yes (machine consist of multiple separately parts)

M_plist	List of the identifiable parts	List of MID	M (limited to 1000 MID), if M_entirety = 1 and M_parts = 1	List of MIDs (parts)

10.11.2 Response Definition – "MachineRegistryResponse"

Dentsu (Name - Type)	**Base Type**	**TPD Type**	**TPD Priority (O = optional, M = Mandatory)**	**Values**
M_ID	string	MID	M	

Status Code	**Description**
201	MachineRegistryResponse
400	List[{ Error_Code: string, Error_Description: string}]
401	VAL_SEC_TOKEN
403	VAL_SEC_CLAIM
500	SYSTEM_ERROR

10.11.3 Specific Message Validations

KEY	**Error Code**	**Error Description**	**http Status**
VAL_SEC_CLAIM	CLAIM_VALIDATION_FAILED	The client has no permission to create a MID	403
VAL_ALREADY_REGISTERED	MID_ALREADY_REGISTERED	The machine serial number {M_NUMBER} is already registered in the system	409
VAL_NOT_REGISTERED	INVALID_MID_CODE	The MID is wrong or does not exist	400

VAL_MP_ALREADY_REGISTERED	FAILED_VALIDATION	The machine {MID} is part of another machine	409

10.12 GI ID Issuer API - Get Machine (GMA)

Use this method to retrieve information related to a Machine ID of your Organisation.
GET/machines/{M_ID} Get machine

10.12.1 Request Definition – "GetMachineRequest"

Dentsu (Name - Type)	**Base Type**	**TPD Type**	**TPD Priority (O = optional, M = Mandatory)**	**Values**
M_ID (path)	string	MID	M	

10.12.2 Response Definition - "MachineView"

Dentsu (Name - Type)	**Base Type**	**TPD Priority (O = optional, M = Mandatory)**
M_Active_Status	Machine status (activated/deactivated)	M
EO_ID	EOID	M
F_ID	FID	M
F_Name	string	O
EO_Name1	string	O
M_Capacity	Integer 64	M
M_ID	MID	M
M_Model	string	M

M_Name	string	M
M_Number	string	M
M_Producer	string	M
M_ATD	string	O
PrevMID_B	boolean	O
PrevMID_ID	MID	O
M_entirety	boolean	O
P_Producer	String	O
P_Model	String	O
P_Number	String	O
P_Mobile	boolean	O
P_ATD1	boolean	O
P_ATD2	string	O
P_Description	string	O
M_parts	boolean	O
M_plist	List of the identifiable parts	O

Status Code	**Description**
200	MachineView
400	List[{ Error_Code: string, Error_Description: string}]
401	VAL_SEC_TOKEN
403	VAL_SEC_CLAIM
500	SYSTEM_ERROR

10.12.3 Specific Message Validations

KEY	Error Code	Error Description	http Status
VAL_SEC_CLAIM	CLAIM_VALIDATION_FAILED	The client has no permission to get machine information	403

10.13 GI ID Issuer API - List Machines (LMA)

Use this method to retrieve information for all Machine IDs of your Organisation that meet the query parameters.

GET/machines Get machines list

10.13.1 Query parameters

Name	Type	Priority	Description
M_Active_Status	boolean	O	Select only machines activated/deactivated
EO_ID	EOID	O	Select only machines linked to the given EO
M_Name	String	O	
F_ID	FID	O	Select only machines linked to the given FID
Page	int	O	Page number, default 1
Sort	string	O	Sort the result by the selected option. Check available values for "sort types" EO_ID, F_ID, PRODUCER_ASC, PRODUCER_DESC, MODEL_ASC, MODEL_DESC, NUMBER_ASC, NUMBER_DESC, STATUS

10.13.2 Response Definition – ListOfMachineView

Machine_List*	List<MachineView>
Current_Page	int
Total_Pages	int

Status Code	**Description**
200	ListOfMachineView
400	List[{ Error_Code: string, Error_Description: string}]
401	VAL_SEC_TOKEN
403	VAL_SEC_CLAIM
500	SYSTEM_ERROR

10.13.3 Specific Message Validations

KEY	**Error Code**	**Error Description**	**http Status**
VAL_SEC_CLAIM	CLAIM_VALIDATION_FAILED	The client has no permission to get machine information	403

10.14 GI ID Issuer API - Correct Machine (CMA)

Use this method to correct the information related to a Machine ID of your Organisation

PUT/machines/{M_ID} Edit machine

10.14.1 Request Definition - "MachineEditRequest"

Dentsu (Name - Type)	Base Type	TPD Type	TPD Priority (O = optional, M = Mandatory)	Values
EO_CODE	string	EO_CODE	M (not editable)	
EO_ID	string	EOID	M	
F_ID	string	FID	M	
M_ID (path)	string	MID	M (not editable)	
M_Producer	string	Text(200)	M, if M_entirety = 1	
M_Model	string	Text(200)	M, if M_entirety = 1	
M_Number	string	Text(200)	M, if M_entirety = 1	
M_Capacity	integer	integer	M, if M_entirety = 1	
M_Name	string	Text(500)	M	
M_ATD	string	Text(100)	M, if M_entirety = 1 and M_parts = 0	
PrevMID_B	boolean	boolean	M	0 – No (first time registration) 1 – Yes
PrevMID_ID	MID	MID	M, if PrevMID_B = 1	
M_entirety	boolean	boolean	M	0 – No (machine part) 1 – Yes (machine)
P_Producer	String	Text(200)	M, if M_entirety = 0	
P_Model	String	Text(200)	M, if M_entirety = 0	
P_Number	String	Text(200)	M, if M_entirety = 0	

P_Mobile	boolean	Boolean	M, if M_entirety = 0	0 – No (fixed part) 1 – Yes (mobile part)
P_ATD1	boolean	Boolean	M, if M_entirety = 0	0 – No 1 – Yes
P_ATD2	string	Text(100)	M, if M_entirety = 0 and P_ATD1 = 1	
P_Description	string	Text(500)	O	
M_parts	boolean	boolean	M, if M_entirety = 1	0 – No (not multiple separately parts) 1 – Yes (machine consist of multiple separately parts)
M_plist	List of the identifiable parts	List of MID	M (limited to 1000 MID), if M_entirety = 1 and M_parts = 1	List of MIDs (parts)

10.14.2 Response Definition- "MachineEditResponse"

Status Code	**Description**
204	Machine modified successfully
400	List[{ Error_Code: string, Error_Description: string}]
401	VAL_SEC_TOKEN
403	VAL_SEC_CLAIM
500	SYSTEM_ERROR

10.14.3 Specific Message Validations

KEY	**Error Code**	**Error Description**	**http Status**

VAL_SEC_CLAIM	CLAIM_VALIDATION_FAILED	The client has no permission to modify a Machine	403
VAL_NOT_EDIT	FAILED_VALIDATION	The field {XX} is not editable	400
VAL_NOT_REGISTERED	INVALID_MID_CODE	The MID is wrong or does not exist	400
VAL_MP_ALREADY_REGISTERED	FAILED_VALIDATION	The machine {MID} is part of another machine	409

10.15 GI ID Issuer API - De-registration of Machine (DMA)

Use this method to de-register a Machine ID of your Organisation

DELETE/machines/{M_ID} De-register machine

De-registering a machine will automatically de-register any machine part of type fixed (P_Mobile = 0) associated with it.

10.15.1 Request Definition - "MachineDeregistrationRequest"

KEY	Base Type	Type in TPD	Priority in TPD	Values in TPD
EO_CODE (query param)	string	EO_CODE	M	
EO_ID (query param)	string	EOID	M	
F_ID (query param)	string	FID	M	
M_ID (path)	string	MID	M	

10.15.2 Response Definition – "MachineDeregistrationResponse"

Status Code	Description
204	Machine Deactivated

400	List[{ Error_Code: string, Error_Description: string}]
401	VAL_SEC_TOKEN
403	VAL_SEC_CLAIM
500	SYSTEM_ERROR

10.15.3 Specific Message Validations

KEY	**Error Code**	**Error Description**	**http Status**
VAL_SEC_CLAIM	CLAIM_VALIDATION_FAILED	The client has no permission to deactivate a machine	403

11 CODE ORDERING

11.1 GI ID Issuer API - Create Unit Order (CUO)

Use this method to request unit level UI codes.

The field "Internal_Reference_Number" is optional and if populated with information that information will be shown in the invoice. If no information gets inserted in the "Internal_Reference_Number" field, the Order ID (generated by the system) will be shown in the invoice instead.

Note: Economic Operators must accept the terms and conditions (via the "Accept_Terms" field) for every submitted order request.

NOTE: Text fields (P_OtherType, P_CN, P_Brand and P_SubType_Name) need to respect the ISO8859-15:1999 charset format

POST/orders/unit Create unit order

11.1.1 Request Definition – "CreateUnitOrderRequest"

Dentsu (Name - Type)	**Base Type**	**TPD Type**	**TPD Priority (O = optional, M = Mandatory)**	**Values**
EO_ID	string	EOID	M	
F_ID	string	FID	M	

Process_Type	boolean	Boolean	M	0 – No (only for fully hand made products) 1 – Yes
M_ID	string	MID	M, if Process_Type = 1	
P_Type	int	Integer	M	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- Not Used 11- Novel tobacco product 12 - Other
P_OtherType	string	Text(200)	M, if P_Type = 12 (other tobacco product)	
P_CN	string	Text(200)	O	
P_Weight	decimal		M	
P_Brand	string	Text(200)	M	
TP_ID	string	TPID	M, if Intended_Market is an EU or UK country;	

			Optional otherwise.	
TP_PN	string	PN	M, if Intended_Market is an EU or UK country; Optional otherwise	
Intended_Market	string	Country	M	
Intended_Route1	boolean	Boolean	M	0 – No 1 – Yes
Intended_Route2	string	Country	M, if Intended_Route1 = 1	Check country definition
Import	boolean	Boolean	M	0 – No 1 - Yes
Req_Quantity	int	Integer	M	(Max = 1 million, Min = 1 code)
No_Cancellation	boolean	Boolean	M	0 – No 1 – Yes
Internal_Reference_Number	string	Text(100)	O	
Accept_Terms	boolean	Boolean	M	0 – No 1 – Yes
P_SubType_Exist	Boolean	Boolean	M	0 – No 1 – Yes
P_SubType_Name	string	Text(200)	M, if P_SubType_Exist = 1	
P_units	int	Integer	M, if P_Type = 1 or 2 or 3	

11.1.2 Response Definition – "CreateUnitOrderResponse" – OrderId

Dentsu (Name - Type)	**Base Type**	**TPD Type**	**TPD Priority (O = optional, M = Mandatory)**
Order_ID	string	uuid	M

Status Code	Description
201	CreateUnitOrderResponse
400	List[{ Error_Code: string, Error_Description: string}]
401	VAL_SEC_TOKEN
403	VAL_SEC_CLAIM
500	SYSTEM_ERROR

11.1.3 Specific Message Validations

KEY	Error Code	Error Description	http Status
VAL_SEC_CLAIM	CLAIM_VALIDATION_FAILED	The client has no permission to request codes	403
VAL_REQ_RANGE	INVALID_RANGE	The requested quantity must be between 1 and 1.000.000 codes per Order	400
VAL_ACCEPT_TERMS	ACCEPT_TERMS_REJECTED	Terms must be accepted to request the generation of new codes	400
VAL_FIE_FORMAT	INVALID_INPUT_FORMAT	The field {'field'} contains invalid characters.	400

11.2 GI ID Issuer API - Create Aggregated Order (CAO)

The field "Internal_Reference_Number" is optional and if populated with information that information will be shown in the invoice. If no information gets inserted in the "Internal_Reference_Number" field, the Order ID (generated by the system) will be shown in the invoice instead.

Note: Economic Operators must accept the terms and conditions (via the "Accept_Terms" field) for every submitted order request.

NOTE: Text field (Internal_Reference_Number) needs to respect the ISO8859-15:1999 charset format

POST/orders/aggregated Create aggregated order

11.2.1 Request Definition – "CreateAggregatedOrderRequest"

Dentsu (Name - Type)	**Base Type**	**TPD Type**	**TPD Priority (O = optional, M = Mandatory)**	**Values**
EO_ID	string	EOID	M	
Internal_Reference_Number	string	Text(100)	O	
Req_Quantity	int	Integer	M	Max value (1M), Min value 1
No_Cancellation	boolean	Boolean	M	0 – No 1 – Yes
Accept_Terms	boolean	Boolean	M	0 – No 1 – Yes

11.2.2 Response Definition – "CreateAggegatedOrderResponse"

Dentsu (Name - Type)	**Base Type**	**TPD Type**	**TPD Priority (O = optional, M = Mandatory)**
Order_ID	string	uuid	M

Status Code	**Description**
201	CreateAggegatedOrderResponse
400	List[{ Error_Code: string, Error_Description: string}]
401	VAL_SEC_TOKEN
403	VAL_SEC_CLAIM
500	SYSTEM_ERROR

11.2.3 Specific Message Validations

KEY	Error Code	Error Description	http Status
VAL_SEC_CLAIM	CLAIM_VALIDATION_FAILED	The client has no permission to request codes	403
VAL_REQ_RANGE	INVALID_RANGE	The requested quantity must be between 1 and 1.000.000 codes per Order	400
VAL_ACCEPT_TERMS	ACCEPT_TERMS_REJECTED	Terms must be accepted to request the generation of new codes	400
VAL_FIE_FORMAT	INVALID_INPUT_FORMAT	The field {'field'} contains invalid characters.	400

11.3 GI ID Issuer API - Get Order by Order Id (GOR)

Use this method to list the details of a specific Order ID.

Note that the field O_Status should be used to know whether the codes are available for download. The normal flow is as follows:

1) Place and order using the CUO or CAO methods. You will receive an "order ID" as a response if the request was successful.
2) Use this method (GOR) to read the O_Status until the Order is in status PROCESSED or EXPORTED. Explanation on the statuses:
    a. REQUESTED: Initial status, the codes are being generated.
    b. PROCESSED: The codes are ready for download but have so far not been downloaded.
    c. EXPORTED: The codes are ready for download and have been downloaded at least once (economic operators can download codes of an individual order as often as required).
    d. CANCELLED: In case the order is cancelled by the Economic Operator in the first 24 hours from submitting the order (provided the NoCancellation flag was not set).
3) Call method GCJ (see section 11.6) to download the codes in JSON format or GCC (see section 11.7) to download the codes in CSV format.

GET/orders/{Order_ID} Get order

11.3.1 Request Definition – "GetOrderRequest"

Dentsu (Name - Type)	Base Type	TPD Type	TPD Priority (O = optional, M = Mandatory)
Order_ID	string	uuid	M

11.3.2 Response Definition – "OrderView"

Dentsu (Name - Type)	Base Type	TPD Type	Values
Order_ID	string	UUID	
EO_ID	string	EOID	
EO_Name1	string	Text(500)	
F_ID	string	FID	Not present if O_Type = AGGREGATE
F_Address	string	Text(5000)	
F_Country	string	Text(2)	
F_Name	string	Text(100)	
Process_Type	boolean	Boolean	0 – No (only for fully hand made products) 1 – Yes
M_ID	string	MID	Not present if Process_Type = 1 or O_Type = AGGREGATE
M_Name	string	Text(500)	
P_Type	int	Integer	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- Not used 11 – Novel tobacco product 12 - Other
P_OtherType	string	Text(200)	
P_CN	string	Text(200)	
P_Weight	decimal		
P_Brand	string	Text(200)	
TP_ID	string	TPID	
TP_PN	string	PN	
Intended_Market	string	Country	
Intended_Route1	boolean	Boolean	0 – No 1 – Yes
Intended_Route2	string	Country	
Import	boolean	Boolean	0 – No 1 - Yes
Req_Quantity	int	Integer	M (max 1M)
No_Cancellation	boolean	Boolean	0 – No 1 – Yes
Creation_Date	string	Date	Order creation Date
Cancellation_Target_Date	string	Date	It must contain the value of the order creation date + 24 hours
Internal_Reference_Number	string	Text(100)	Must be unique per organisation
O_Status	string	Text(20)	REQUESTED, PROCESSED, EXPORTED, CANCELLED
O_Type	string	Text(20)	UNIT, AGGREGATED
Accept_Terms	boolean	Boolean	
P_SubType_Exist	Boolean	Boolean	0 – No 1 – Yes
P_SubType_Name	string	Text(200)	M, if P_SubType_Exist = 1
P_units	int	Integer	M, if P_Type = 1 or 2 or 3

ErrorDetail:

Dentsu (Name - Type)	Base Type	TPD Type	Values
Error_Code	string	Text	
Error_Descr	string	Text	

Status Code	Description
200	GetOrderResponse
400	List[{ Error_Code: string, Error_Description: string}]
401	VAL_SEC_TOKEN
403	VAL_SEC_CLAIM
500	SYSTEM_ERROR

11.3.3 Specific Message Validations

KEY	Error Code	Error Description	http Status
VAL_SEC_CLAIM	CLAIM_VALIDATION_FAILED	The client has no permission to get order details	403

11.4 GI ID Issuer API - Get Order List (GOL)

Use this method to retrieve information related to all orders previously submitted for your organisation and which meet the query parameter criteria.

GET/orders Get order list

11.4.1 Query parameters

Name	Type	Priority	Description

EO_ID	EOID	O	Select orders with given EO ID
Intended_Market	Array(string)	O	Select intended markets
M_ID	Array(string)	O	List of machine IDs
No_Cancellation	boolean	O	
P_Type	int	O	Select orders with given product types
O_Type	text	O	Select orders with given order type
O_Status	text	O	Select orders with given order status
Internal_Reference_Number	text	O	
Page	int	O	Page number, default 1
Sort	string	O	Sort the result by the selected option. Check available values for "sort types" EO_ID, O_TYPE_ASC, O_TYPE_DESC, O_STATUS_ASC, O_STATUS_DESC, P_TYPE_ASC, P_TYPE_DESC, M_ID_ASC, M_ID_DESC, INTENDED_MARKET_ASC, INTENDED_MARKET_DESC

11.4.2 Response Definition – "ListOfOrderView"

Order_List*	List<OrderView>
Current_Page	int
Total_Pages	int

Status Code	Description
200	ListOfOrderView
400	List[{ Error_Code: string, Error_Description: string}]
401	VAL_SEC_TOKEN
403	VAL_SEC_CLAIM
500	SYSTEM_ERROR

11.4.3 Specific Message Validations

KEY	Error Code	Error Description	http Status
VAL_SEC_CLAIM	CLAIM_VALIDATION_FAILED	The client has no permission to get order details	403

11.5 GI ID Issuer API - Cancel Order Id (COI)

Use this method to cancel a submitted order. This option is only available if the flag NoCancellation was not set during the placing of an order.

DELETE/orders/{Order_ID} Cancel order

11.5.1 Request Definition – "CancelOrderRequest"

Dentsu (Name - Type)	Base Type	TPD Type	TPD Priority (O = optional, M = Mandatory)
Order_ID (path)	string	uuid	M

11.5.2 Response Definition – "CancelOrderResponse"

Status Code	Description
204	Order cancelled
400	List[{ Error_Code: string, Error_Description: string}]

401	VAL_SEC_TOKEN
403	VAL_SEC_CLAIM
500	SYSTEM_ERROR

11.5.3 Specific Message Validations

KEY	**Error Code**	**Error Description**	**http Status**
VAL_SEC_CLAIM	CLAIM_VALIDATION_FAILED	The client has no permission to cancel an order	403
VAL_CANCEL_24H	CANCELLATION_WITHIN_24_HOURS	Cancellation event should be performed within 24 hours of the order creation	400
VAL_CANCEL_ PROCESSED	ORDER_ALREADY_PROCESSED	The order {Order_ID} is already processed and can't be cancelled	400
VAL_NOT_CANCEL	ORDER_NOT_CANCELLABLE	The order {Order_ID} can't be cancelled because it was created with the parameter "No_Cancellation" = true	400

11.6 GI ID Issuer API - Get generated codes by Order Id (JSON) (GCJ)

Use this method to retrieve the codes generated in JSON format once the order is in status PROCESSED or EXPORTED. The order will be presented in several JSON pages defined by the "Total_Pages" parameter. You can fetch any page at any time by using the "Current_page" parameter.

GET/orders/{Order_ID}/codes Get generated codes JSON

11.6.1 Query parameters

Dentsu (Name - Type)	Base Type	TPD Type	TPD Priority (O = optional, M = Mandatory)
Order_ID (path)	string	uuid	M
Page (path)	int	Integer	M, default = 1

11.6.2 Response Definition – "JSONOrderView"

Dentsu (Name - Type)	Base Type	TPD Type	TPD Priority (O = optional, M = Mandatory)
Codes	Array(String)	upUI(i)/aUI	M
Codes_upUI_s	Array(String)	Human Readable - upUI(s)	M, if Codes=upUI(i)
Current_Page	int	Integer	M
Total_Pages	int	Integer	M

Status Code	Description
200	JSONOrderView
400	List[{ Error_Code: string, Error_Description: string}]
401	VAL_SEC_TOKEN
403	VAL_SEC_CLAIM
500	SYSTEM_ERROR

11.6.3 Specific Message Validations

KEY	Error Code	Error Description	http Status
VAL_SEC_CLAIM	CLAIM_VALIDATION_FAILED	The client has no permission to get an order	403

VAL_PAGE_NOT_FOUND	PAGE_OVERFLOW	The requested page does not exist	400
VAL_ORDER_STATUS	ORDER_INVALID_STATUS	The order is not available to download yet	400

11.7 GI ID Issuer API - Get generated codes by Order ID (CSV) (GCC)

Use this method to retrieve codes using the CSV format. The response will include a link pointing to the download location of the CSV file.

GET/orders/{Order_ID}/file Get generated codes file link

11.7.1 Query parameters

Dentsu (Name - Type)	**Base Type**	**TPD Type**	**TPD Priority (O = optional, M = Mandatory)**
Order_ID (path)	string	uuid	M

11.7.2 Response Definition – "CSVOrderView"

Dentsu (Name - Type)	**Base Type**	**TPD Type**	**TPD Priority (O = optional, M = Mandatory)**
File_URL	string	Text	M

Status Code	**Description**
200	CSVOrderView
400	List[{ Error_Code: string, Error_Description: string}]
401	VAL_SEC_TOKEN
403	VAL_SEC_CLAIM
500	SYSTEM_ERROR

11.7.3 Specific Message Validations

KEY	Error Code	Error Description	http Status
VAL_SEC_CLAIM	CLAIM_VALIDATION_FAILED	The client has no permission to get an order	403

11.7.4 Clarification on CSV code format for download

The CSV will contain the 2 different formats of the codes in the same line separated with **#.** This is **upUI(i)#upUI(s)**

12 OTHER FUNCTIONALITIES

12.1 GI ID Issuer API - Entity Verification (ICV)

Use this method to verify any Economic Operator ID, Facility ID or Machine IDs stored in the GI Track & Trace system, including identifier codes that belong to other organisations.

This verification method returns information on the three activities listed below but does not disclose any information contained in the identifier codes as such (e.g., address):

- Existence of EOID / FID / MID;
- Validation status (Active / Inactive) of EOID / FID / MID;
- Relationship between EOID – FID or FID – MID.

POST/icv/lookup Verify entities

12.1.1 Request Definition – "EntityVerificationRequest"

KEY	Value(Type)	TPD Description	Priority in TPD	Values in TPD
EO_IDS	Array(string)	EOID	O	
F_IDS	Array(string)	FID	O	
M_IDS	Array(string)	MID	O	
R_EOF	array of object<string>	A list of relation of EOID and FID to check for existence	O	Example: List<{EOID123, FID123}> (Query means: is

				FID123 children of EOID123?)
R_EOFM	array of object <strings>	A list of relation of EOID, FID and MID to check for existence	O	Example: List<{EOID123, FID123, MID123}> (Query means: is FID123 children of EOID123 and MID123 child of FID123?)

12.1.2 Response Definition – "EntityVerificationResponse"

Dentsu (Name - Type)	Base Type	TPD Type	TPD Priority (O = optional, M = Mandatory)
EO_IDS_VALID	Array(bool)	Boolean	M
F_IDS_VALID	Array(bool)	Boolean	M
M_IDS_VALID	Array(bool)	Boolean	M
R_EOFM_VALID	Array(bool)	Boolean	M
R_EOF_VALID	Array(bool)	Boolean	M

12.1.3 Response example

```
{
 "EO_IDS_VALID": [
  0,1,0,1
 ],
 "F_IDS_VALID": [
  0,0,0
 ],
 "M_IDS_VALID": [
  0
 ],
 "R_EOFM_VALID": [
  0,1,1
 ],
 "R_EOF_VALID": [
  0,0,0,1,1
 ]}
```

The true/false (1/0) responses are mapped as follows:

- If the EOID/FID/MID **exists** and is **active,** then "1";
- If the EOID/FID/MID **exists** and is **inactive,** then "0";
- If the EOID/FID/MID **does not exist**, then "0".

For the relationships, the method ignores the activated/inactivated status:

- If any of the EOID/FID/MID in the relationship does not exist, then "0"
- If any of the parent-child relationships do not exist, then "0"
- Otherwise = "1"

Status Code	**Description**
200	EntityVerificationResponse
400	List[{ Error_Code: string, Error_Description: string}]
401	VAL_SEC_TOKEN
403	VAL_SEC_CLAIM
500	SYSTEM_ERROR

12.1.4 Specific Message Validations

KEY	**Error Code**	**Error Description**	**http Status**
VAL_SEC_CLAIM	CLAIM_VALIDATION_FAILED	The client has no permission to verify entities	403

13 Endpoints

The following URLs are the system URLs.

Service	**PRE-PROD URL (Public Testing/Integration environment)**
GI ID Issuer API	https://api.idissuer.pre.gitobaccotracing.com
GI ID Issuer API (Swagger)	https://api.idissuer.pre.gitobaccotracing.com/swagger/index.html
GI ID Issuer API (Auth)	https://auth.pre.gitobaccotracing.com/oauth2/token
GI ID Issuer Portal	https://idissuer.pre.gitobaccotracing.com
GI Gateway (JSON)	https://api.gateway.pre.gitobaccotracing.com
GI Gateway (Auth)	https://auth.pre.gitobaccotracing.com
Service	**PROD URL (Production URL)**
GI ID Issuer API	https://api.idissuer.prod.gitobaccotracing.com
GI ID Issuer API (Swagger)	https://api.idissuer.prod.gitobaccotracing.com/swagger/index.html
GI ID Issuer API (Auth)	https://auth.prod.gitobaccotracing.com/oauth2/token
GI ID Issuer Portal	https://idissuer.prod.gitobaccotracing.com
GI Gateway (JSON)	https://api.gateway.prod.gitobaccotracing.com
GI Gateway (Auth)	https://auth.prod.gitobaccotracing.com
Document Center	https://www.gitobaccotracing.com/
Support Portal	https://support.gitobaccotracing.com/giportal

14 List of Standards

1	OAuth 2	https://www.oauth.com/oauth2-servers/access-tokens/client-credentials/
2	ISO/IEC 9834-8:2014	https://www.iso.org/standard/62795.html

	Information technology -- Procedures for the operation of object identifier registration authorities -- Part 8: Generation of universally Unique identifier (UI) (UUIs) and their use in object identifiers	
3	The JavaScript Object Notation (JSON) Data Interchange Format Internet Engineering Task Force (IETF) Request for Comments: 8259	https://tools.ietf.org/html/rfc8259
4	UUID	https://www.ietf.org/rfc/rfc4122.txt