



eLot and Tracking Unit Register Web API Guide

Version Date: 03 August 2020

Table of Contents

Version Date: 03 August 2020.....	1
Introduction	3
Web API Versions	3
Pre-requisites	4
How to Request API Access on Central Business Register (CBR).....	4
How to Connect.....	4
Testing Web API End Points	4
Swagger.....	4
Available Endpoints	5
eLot (POST).....	5
Get Tracking Unit Status (GET)	9
Tracking Unit Statuses (POST)	10
Get Tracking Unit by Verification Key (GET).....	11
Get Tracking Units By eLot Key (GET)	13
Add Inspection Result (POST).....	14
Update Inspection Result (POST)	15
Tracking Unit (GET).....	16

Introduction

The Tracking Unit Register (TUR) has been introduced as a central database in the official export certification system to store relevant data around tracking units (e.g. in the fruit industry this means data relevant for pallets). This TUR concept replaces paper-based evidence to demonstrate compliance with the export conditions, as users can now update and call upon the TUR to provide the required confirmation.

The eLot step has been introduced in the certification process and was included for two main purposes. Firstly, the eLot is a means for tracking unit data to be inserted into the TUR. Secondly, once the data is received by the TUR, the process of pre-verification can take place to ensure the product has met the requirements (up to that point) to be able to be considered for certification. The objective is to prevent defective product from entering the supply chain right at the start of the supply chain.

This guide discusses the eLot API, which is the web service to submit data directly to the TUR and how to interpret the response received back from this service. Clients will also be shown how to call for information from the TUR so they can track the status of tracking units. The first sections of this document are concerned with how to register and connect to these services, before going into more detail on each of the API endpoints.

Web API Versions

There are three versions of the API that are presently available at the time and these are stated below:

- **Old**- Soon to be deprecated
- **Stable** – Current stable version
- **Beta**-Latest Build

All requests are sent with the version of the API, the consumer needs to call through a custom request header as below:

```
Accept-version: v1  
Accept-version: v2
```

The current version of the API will be displayed on the API's Swagger page. The swagger page will display the available endpoints depending on the selected version.

When a new version becomes available, notifications will be sent out to all the relevant parties with a changelog of the new changes and features. The Swagger page will be updated accordingly with these new features.

The links to the swagger pages are as follows:

- PRODUCTION - <https://tur.ecert.co.za/swagger>
- TESTING - <http://qa.tur.ecert.co.za/swagger>

Pre-requisites

The pre-conditions for a consuming (client) system to be able to use the TUR Web API are as follows:

- The consuming system must belong or be associated with a registered business in the Central Business Register (<https://cbr.ecert.co.za/>)
- A client system calling the TUR system will need to be registered as a client in the User Authentication Service(UAS) and issued a valid client Id and client secret to be able to consume the endpoints ([see section 1 below](#))
- All request calls to the TUR will be authenticated using a JWT (JSON Web token) bearer token over OAuth 2.0 protocol (see [How To Connect](#) section)

How to Request API Access on Central Business Register (CBR)

Please refer to the **Request API Access** user guide.

Once a Client has been registered successfully a ClientId, **ClientSecret** will be generated. The **ClientId** and **ClientSecret**, in turn, will be used by the client to authorize and get a token. With a valid token, the client system can call any API endpoint on the TUR Web API provided all the required parameters are presented.

How to Connect

Refer to the **How to Connect to Web API Document**

Testing Web API Endpoints

To test the TUR Web API endpoints users can use 3rd party tools like Swagger or Postman. These tools will help the users to get a feel of what the endpoint parameters look like as well as view the responses in different formats. Below are screenshots to show how to test using both of the tools:

Swagger

- <http://qa.tur.ecert.co.za/swagger> (TESTING)

Available Endpoints

eLot (POST)

This eLot Notice web service must be used to submit tracking unit data to the TUR. Pre-verification of the tracking units takes place and the pre-verification results are returned to the client. A brief description of how the key data elements are provided and how to interpret the response is discussed in the table below.

The decision on when to submit the data depends on the client but must be before PPECB quality inspections. Practically data should be submitted when there is certainty regarding the completeness of the tracking unit composition (i.e. it is unlikely to physically change) and the destination of that product is known. The destination is important because the pre-verification process will run the appropriate rules linked to that intended market. The intended market is communicated using the AgreementCode element – which is linked to the Agreements eCertification standard.

IsTest: Client can also trigger the “IsTest” flag as a means to assess the suitability of that product for a possible market without committing to it. The API will run in the same way and appropriate information provided back in the response, but the tracking unit data or results will not be stored in the TUR. If clients want to select that market based on favourable test results the eLot will have to be re-submitted so that the data is added to the TUR and the pre-verification process can be formally undertaken and the results stored.

AgreementCode: The AgreementCode can be included at the header level or the Tracking Unit (TU) level. The AgreementCode is optional at the Tracking Unit level and if it is not passed it uses the same value as the one passed on the header. If the AgreementCode is set on the TU level, then it will override the value passed on the header. Having the AgreementCode at the TU level allows for different Agreement Codes to be set against tracking units as opposed to one Agreement Code being allocated to the eLot.

Please note: An AgreementCode has to be specified at the header or TU level otherwise the database will throw a validation error as this is a required field in the database.

TrackingUnitID: Please note that for the fruit industry, it has been agreed that the tracking unit identification (i.e. the pallet id) must be in the GS1 format (i.e. SSCC). Tracking Unit identification that does not use this convention will not be successfully pre-verified which will result in the product not being certified. It is expected other sectors will also likely use the GS1 standards.

Reference1: Is provided by the client. This Reference has been reserved for the Consignment Note information in the fruit sector if it is known at eLot stage. This may be useful to track consignment of tracking units later.

Reference2: Is provided by the client. In the eLot API, this is an optional reference the client can provide. Examples might be to flag pallets for other parties in their supply chain.

ExportDate: Is provided as the intended date of export. This is relevant when new phytosanitary rules may apply and the system needs to know when the product will be exported so the correct rules are

used for pre-verifying. Practically, the current date can be used or the typical timeframe between packing and departure of the product.

Endpoint URL: <http://qa.tur.ecert.co.za/api/TrackingUnit/eLot>

Request Parameters

Parameter Name	Description	Req?	Data Type	Example
BusinessID	Business identification from CBR database	Yes	String	4
Industry	Reference to which industry this TU is from	Yes	String	FRUIT
AgreementCode	Agreement code used for the export market	No	String	AGM0014
IsTest	Whether or not this is a test call	Yes	Boolean	False. (Default = true)
TrackingUnits	<i>A list of tracking units (including tracking unit detail) to add/update to the system and pre-verify</i>	Yes	Array	
TrackingUnitID	Tracking Unit Identifiers (GS1 Format - SSCC)	Yes	String	e.g. 660091600071233222
AgreementCode	Agreement code used for the export market	No	String	AGM0014
Reference1	Provided by client system e.g. consignment number for tracking	No	String	
Reference2	Providing by client system e.g. other flags to identify tracking unit	No	String	
ExportDate	Intended Date of Export	No	DateTime	e.g. 2020-01-31
Weight	Weight of tracking unit	Yes	Decimal	1800.30
WeightUnitCode	Unit of Measure (ISO Code - 2)	Yes	String	KG
NumberOfPackageItems	Item count of packaged items	Yes	Int	20
TrackingUnitDetails	<i>A list of tracking unit details that are contained on a single tracking unit</i>	Yes	Array	
OperatorCode	Production Unit Code – (PUC)		String	e.g. D1234

OriginLocation	Orchard		String	e.g. K12
SPSStatus	Phyto Status Field		String	
PackOperatorCode	Packhouse Code (PHC)		String	e.g. L4352
CommodityCode	Commodity Code (2 Character Code)	Yes	String	e.g. LE (<i>LEMONS</i>)
MarketingIndicationCode	Variety Code	Yes	String	e.g. EUR (<i>EUREKA</i>)
ClassCategory	Product class category	Yes	String	I or II or P
NumberOfPackagedItems	Item count of packaged items for example cartons	Yes	Int	23
PackageType	Type of packaging (ISO Code - 2)	Yes	String	CT
TrackingUnitLocation	Current location of tracking unit (<i>Future Purposes</i>)	No	String	
TrackingUnitOrigin	Original province the tracking unit was created from (Province Code)	Yes	String	GP
InspectionPoint	Location Code where tracking units will be inspected	Yes	String	1024

Response

Property Name
IsSuccessful
eLotKey
Message
Data (array) <ul style="list-style-type: none"> TrackingUnitID VerificationKey ProcessStatus ProcessResult (array) RejectionReasons (array) TrackingUnitDetails(array)

This section deals with a basic interpretation of the API response.

eLotKey is a unique TUR system generated reference linked to this API call for this set of tracking units.

Message is an array of information returned as a summary of the information across all the tracking units in this eLot. An example might be identifying the pallet with the “worst” phytosanitary status – flagged as the determining status for the whole group of tracking units.

VerificationKey is a unique system-generated reference for this specific tracking unit for this process.

ProcessStatus is the summary result of the process for this particular tracking unit. The possible status values for each process will be reflected in the eCertification standards, but in most cases will be either "Pending", "Passed" or "Failed".

ProcessResult is an array of information per tracking unit reflecting the performance against the underlying export conditions being evaluated. Possible values in this array will depend on the process and the underlying export conditions and will correspond with the eCertification standard for Process Results. Conceptually, this field is intended to replace, or be equivalent to, the existing "PhytoData" information. The pallet is thus obtaining a certain phytosanitary status which then remains with that pallet.

RejectionReasons is an array of reasons why the tracking unit has been found not to meet the underlying export condition requirements. The list of reasons corresponds to the eCertification standard for Rejection Reasons per process per agreement.

Examples of Rejection Reasons in the eLot process:

- Variety Non-compliance
- Marketing Indications Non-compliance
- FBO PUC registration Non-compliance
- FBO PHC registration Non-compliance
- Orchard registration Non-compliance
- PUC count per pallet Non-compliance
- FBO not recognized
- Tracking Unit Identifier Non-compliance

Example eLot Response:

```
{
  IsSuccessful: true,
  eLotKey: "4BA1986",
  Message: ["Message1", "Message2"],
  Data: [{
    TrackingUnitID: "760016510576306003",
    VerificationKey: "4CA1556",
    ProcessStatus: "Passed",
    ProcessResult: "EUA1A1FY",
    RejectionReasons: []
  }, {
    TrackingUnitID: "860016510576596003",
    VerificationKey: "5RA1534",
    ProcessStatus: "Failed",
    ProcessResult: "0",
    RejectionReasons: ["FBO PUC registration Non-compliance"]
  }
}
```



```
]
}
```

Get Tracking Unit Status (GET)

Get status of Tracking Units for each export process. If the results are required for specific export processes, then the ExportProcess element can be used to specifying the export processes required.

Endpoint URL: <http://qa.tur.ecert.co.za/api/TrackingUnit/GetTrackingUnitStatus>

Request Parameters

Parameter Name	Description	Required?	Data Type	Example
TrackingUnitID	Identify TU in the Tracking Unit Register	Yes	string	[760016510576306003, 860016510576596003]
ExportProcess		No	string	[eLot, eCert, eInspect] eLot = Packhouse Pre-Verification eCert = Preloading Check eInspect = Packhouse Phytosanitary Inspection

Response

Property Name	
TrackingUnitID	
Industry	
BusinessID	
AgreementCode	
Weight	
WeightUnitCode	
NumberOfPackageItems	
Reference1	
Reference2	
ExportDate	
RegimeCode	
LoadPort	
VerificationKey	
TrackingUnitStatuses (array)	<ul style="list-style-type: none"> • ExportProcess (string) • ProcessStatus (string) • ProcessResult(array) • RejectionReasons(array)

	<ul style="list-style-type: none"> • Reference1(string) • Reference2(string) • UpdatedDateTime(datetime)
TrackingUnitDetails (array)	<ul style="list-style-type: none"> • OperatorCode (string) • OriginLocation (string) • SPSSStatus (string) • PackOperatorCode (string) • CommodityCode (string) • MarketingIndicationCode(String) • ClassCategory (string) • NumberOfPackagelItems (integer) • PackageType (string) • TrackingUnitLocation (string) • TrackingUnitOrigin (string) • InspectionPoint(string)

Tracking Unit Statuses (POST)

Get status of Tracking Unit for each export process

Endpoint URL: <http://qa.tur.ecert.co.za/api/TrackingUnit/GetTrackingUnitStatuses>

Request Parameters

Parameter Name	Description	Required ?	Data Type	Example
TrackingUnitIds	Array of Tracking Unit IDs	Yes	array	["12345678989987654321","123456789987654322"]
ExportProcesses		Yes	string	PhytosanitaryInspection

Response

This endpoint returns an array of the Tracking Units

Property Name
TrackingUnitID
Industry
BusinessID
AgreementCode
Weight
WeightUnitCode
NumberOfPackagelItems
Reference1
Reference2
ExportDate

RegimeCode	
LoadPort	
VerificationKey	
TrackingUnitStatuses (array)	<ul style="list-style-type: none"> • ExportProcess (string) • ProcessStatus (string) • ProcessResult(array) • RejectionReasons(array) • Reference1(string) • Reference2(string) • UpdatedDateTime(datetime)
TrackingUnitDetails (array)	<ul style="list-style-type: none"> • OperatorCode (string) • OriginLocation (string) • SPSStatus (string) • PackOperatorCode (string) • CommodityCode (string) • MarketingIndicationCode(String) • ClassCategory (string) • NumberOfPackageItems (integer) • PackageType (string) • TrackingUnitLocation (string) • TrackingUnitOrigin (string) • InspectionPoint(string)

Get Tracking Unit by Verification Key (GET)

Get status of a Tracking Unit by using the returned VerificationKey.

Endpoint URL: <http://qa.tur.ecert.co.za/api/TrackingUnit/GetTrackingUnitByVerificationKey>

Request Parameters

Parameter Name	Description	Required?	Data Type	Example
VerificationKey	Verification key of the TU in the system	Yes	string	4CA1556
ExportProcess		No	array	[eLot, eCert, eInspect]

Response

Property Name
TrackingUnitID
Industry
BusinessID
AgreementCode
Weight
WeightUnitCode

NumberOfPackageltems	
Reference1	
Reference2	
ExportDate	
RegimeCode	
LoadPort	
VerificationKey	
TrackingUnitStatuses (array)	<ul style="list-style-type: none"> • ExportProcess (string) • ProcessStatus (string) • ProcessResult(array) • RejectionReasons(array) • Reference1(string) • Reference2(string) • UpdatedDateTime(datetime)
TrackingUnitDetails (array)	<ul style="list-style-type: none"> • OperatorCode (string) • OriginLocation (string) • SPSStatus (string) • PackOperatorCode (string) • CommodityCode (string) • MarketingIndicationCode(String) • ClassCategory (string) • NumberOfPackageltems (integer) • PackageType (string) • TrackingUnitLocation (string) • TrackingUnitOrigin (string) • InspectionPoint(string)

Get Tracking Units By eLot Key (GET)

Get status of a Tracking Units

Endpoint URL: <http://qa.tur.ecert.co.za/api/TrackingUnit/GetTrackingUnitByeLotKey>

Request Parameters

Parameter Name	Description	Required?	DataType	Example
eLotKey	eLot key of the tracking units in the system	Yes	string	4CA1556
ExportProcess		No	array	[eLot, eCert, eInspect]

Response

Property Name	
TrackingUnitID	
Industry	
BusinessID	
AgreementCode	
Weight	
WeightUnitCode	
NumberOfPackageltems	
Reference1	
Reference2	
ExportDate	
RegimeCode	
LoadPort	
VerificationKey	
TrackingUnitStatuses (array)	<ul style="list-style-type: none"> • ExportProcess (string) • ProcessStatus (string) • ProcessResult(array) • RejectionReasons(array) • Reference1(string) • Reference2(string) • UpdatedDateTime(datetime)
TrackingUnitDetails (array)	<ul style="list-style-type: none"> • OperatorCode (string) • OriginLocation (string) • SPSStatus (string) • PackOperatorCode (string) • CommodityCode (string) • MarketingIndicationCode(String) • ClassCategory (string) • NumberOfPackageltems (integer) • PackageType (string)

- TrackingUnitLocation (string)
- TrackingUnitOrigin (string)
- InspectionPoint(string)

Add Inspection Result (POST)

This endpoint is relevant for official inspection bodies that can provide feedback on the inspection status of tracking units. For example, this endpoint is intended to be used by PPECB to submit PHC Quality inspection results. Permissions will apply to ensure that only the correct institutions are adding or updating inspection results.

Endpoint URL: <http://qa.tur.ecert.co.za/api/TrackingUnit/AddInspectionResult>

Request Parameters

Parameter Name	Description	Required?	Data Type	Example
<i>Inspections</i>	<i>Compose of a list of results for each tracking unit</i>	<i>Yes</i>	<i>Array</i>	
TrackingUnitID	Identify TU in the Tracking Unit Register	Yes	string	860016510576596003
Industry	Reference to which industry this TU is from	Yes	string	FRUIT
ExportProcess	What process the tracking unit is being involved in	Yes	string	eCert
ProcessStatus	An indication of the status for the Agreement	Yes	string	Passed
ProcessResult	The outcome of the process	Yes	string	80
RejectionReasons	The reason the process failed	Yes	array	Inappropriate Cartons
BusinessID	Business identification from CBR database	Yes	string	4
Reference1	Client Reference 1	No	string	XYZ123
Reference2	Client Reference 2	No	string	CL1234

Response

Property Name
IsSuccessful
Message

Update Inspection Result (POST)

This endpoint is relevant for official inspection bodies that can provide feedback on the inspection status of tracking units. For example, this endpoint is intended to be used by PPECB to submit PHC Quality inspection results. Permissions will apply to ensure that only the correct institutions are adding or updating inspection results.

Endpoint URL: <http://qa.tur.ecert.co.za/api/TrackingUnit/UpdateInspectionResult>

Request Parameters

Parameter Name	Description	Required?	Data Type	Example
<i>Inspections</i>	<i>Compose of a list of results for each tracking unit</i>	<i>Yes</i>	<i>Array</i>	
TrackingUnitID	Identify TU in the Tracking Unit Register	Yes	string	860016510576596003
Industry	Reference to which industry this TU is from	Yes	string	FRUIT
ExportProcess	What process the tracking unit is being involved in	Yes	string	eCert
ProcessStatus	An indication of the status for the Agreement	Yes	string	Passed
ProcessResult	The outcome of the process	Yes	string	80
RejectionReasons	The reason the process failed	Yes	array	Inappropriate Cartons
BusinessID	Business identification from CBR database	Yes	string	4
Reference1	Client Reference 1	No	string	XYZ123

Reference2	Client Reference 2	No	string	CL1234
------------	--------------------	----	--------	--------

Response

Property Name
IsSuccessful
Message

Tracking Unit (GET)

Get tracking unit

Endpoint URL: <http://qa.tur.ecert.co.za/api/TrackingUnit/GetTrackingUnit>

Request Parameters

Parameter Name	Description	Required?	Data Type	Example
TrackingUnitID		Yes	string	12345678989987654321

Response

Property Name	
TrackingUnitID	
Reference1	
Reference2	
ExportDate	
Weight	
WeightUnitCode	
NumberOfPackageItems	
TrackingUnitDetails (array)	<ul style="list-style-type: none"> • OperatorCode (string) • OriginLocation (string) • SPSSStatus (string) • PackOperatorCode (string) • CommodityCode (string) • MarketingIndicationCode(String) • ClassCategory (string) • NumberOfPackageItems (integer) • PackageType (string) • TrackingUnitLocation (string) • TrackingUnitOrigin (string) • InspectionPoint(string)