LIV|EX

THE FINE WINE MARKET

Commodity Code API v1

Document Revision 1.2
Date of Issue: 04 January 2021
Date of revision: 5 October 2021

Daria Ershova

Product Manager

Table of Contents

1. Purpose

To provide the API end point information and examples of the web services available for Commodity Code.

2. Glossary of Terms

Term	Meaning
Commodity code (UK) **TARIC (EU)**	A commodity code is a sequence of numbers made up of six, eight or ten digits which are used to determine: - the customs authority duties and other charges that may be levied on the specific goods - The restrictions and prohibitions that may apply to the import, export or transit of the goods. The six-digit commodity codes are known as the HS codes. They are used worldwide in monitoring trade volumes and applying international trade measures to goods.
LWIN	LWIN - the Liv-ex Wine Identification Number – serves as a universal wine identifier for the wine trade. LWIN is a unique seven to eighteen-digit numerical code that can be used to quickly and accurately identify a product. LWIN allows wine companies to keep their preferred naming system, while introducing a new universal code.

3. Technical Standards

- Permitted users will be issued with a unique token (CLIENT_KEY) and password (CLIENT_SECRET) combination to control the access for all the web services covered under Exchange Integration.
- The web services will consume and produce both XML and JSON. The user can provide the content type in the request header. If the user does not provide any information, then the default content type will be JSON.
- The service supports ISO 8601.
- The service only support HTTPS protocol for client and server communications.
- The API will support the following methods:
    - POST for read operation
- Pretty printing for output readability only is supported if required
- Compression for bandwidth savings are used
- Authentication mechanism will be custom based on CLIENT_KEY and CLIENT_SECRET
- The APIs will be accessible at https://api.liv-ex.com/ followed by their specific base URIs

4. Request Header

This information will be used to authenticate valid access to the REST API. Each user will have to provide the following information in the request header. Please note that the API expects the 4 headers as listed within this documentation and submitting a request with additional headers may lead to errors and/or failed responses.

Parameter

Name	Mandatory	Description
CLIENT_KEY	Y	A valid GUID which will be unique for each user.
CLIENT_SECRET	Y	Password/Secret for the merchants CLIENT_KEY.
ACCEPT	Y	Accept header is a way for a client to specify the media type of the response content it is expecting. The values for the content type will be application/json or application/xml. If no/ invalid content type is found in the request, then JSON format will be used by default.
CONTENT-TYPE	Y for POST requests	Content-type is a way to specify the media type of request being sent from the client to the server. The values for the content type will be application/json or application/xml. If no/ invalid content type is found in the request, then JSON format will be used by default.

Example header

```
CLIENT_KEY: 12A34BC56-DE7F-89G0-H1J2345K678L
CLIENT_SECRET: dummy_password
ACCEPT: application/json
CONTENT-TYPE: application/json
```

Invalid header (JSON response)

```
{
    "status": "Unauthorized",
    "httpCode": "401",
    "message": "Unauthorized",
    "internalErrorCode": null,
    "apiInfo": {
        "version": "1.0",
        "timestamp": 1554364615297,
        "provider": "Liv-ex"
    }
}
```

Invalid header (XML response)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Response xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="https://aby-qa-api.liv-ex.com/v1 https://aby-qa-api.liv-
ex.com/schema/v1/services.xsd">
    <Status>Unauthorized</Status>
    <HttpCode>401</HttpCode>
    <Message>Unauthorized</Message>
    <InternalErrorCode xsi:nil="true"/>
    <ApiInfo>
        <Version>1.0</Version>
        <Timestamp>2019-04-04T12:02:37.092+01:00</Timestamp>
        <Provider>Liv-ex</Provider>
    </ApiInfo>
```

```
</Response>
```

5. API Listing

5.1 Commodity code service (POST method)

Description

This service allows users to request a commodity code for a specified product, vintage and bottle size combination (LWIN16/18).

Commodity codes can be generated for the following types of products (LWIN Type): wine, fortified wines, spirits.

There are three available Commodity code standards:

1. United Kingdom: https://www.trade-tariff.service.gov.uk/chapters/22 (headings 2204 and 2208)
   - Wine, Fortified wine – up to 10 digits
   - Spirits – up to 6 digits
2. European Union: https://ec.europa.eu/taxation_customs/dds2/taric/taric_consultation.jsp?Lang=en (headings 2204 and 2208)
   - Wine, Fortified wine – up to 10 digits
   - Spirits – up to 6 digits
3. Singapore (and ASEAN member countries): https://www.customs.gov.sg/documents/businesses/stcced-2018-apr-20.pdf (headings 2204 and 2208)
   - Wine, Fortified wine – up to 8 digits
   - Spirits – up to 8 digits

Commodity code structure:

Chapter in the Harmonized System (HS)	2 digits	e.g. '**Chapter 22** Beverages, spirits and vinegar'
HS heading	4 digits	e.g. '**2204** Wine of fresh grapes, including fortified wines; grape must other than that of heading 2009'
HS subheading	6 digits	e.g. '**2204 10** Sparkling wine'
CN subheading	8 digits	e.g. '**2204 10 11** With a protected designation of origin (PDO)'
TARIC subheading	10 digits	e.g. '**2204 10 11 00** Champagne'

Please note that codes are not available for the following products (for a detailed list of exclusions please see Section 7 "Excluded codes"):

- Wines with 'mushroom' stoppers held in place by ties or fastenings; wine, otherwise put up, with an excess pressure due to carbon dioxide in solution of not less than 1 bar but less than 3 bar.
- Wines of *not* fresh grapes
- Grape must

Base URI

/data/v1/commodityCode

Request Parameters

Name	Mandatory	Description
lwin	Y	LWIN16/18 code. Only one value is permitted per request **Type:** 16/18-digit integer **Example**: 1011872199500750, 101187219951200750
commodityCodeType	Y	Required commodity code standard **Values:** UK, EU, SG **Type:** alphanumeric
alcoholValue	N	Alcohol value for the given product and vintage. **Type:** double

Sample Request Body

JSON Request

Using Liv-ex alcohol value

```
{
    "commodityCode":{
    "lwin":"112266220160600750",
    "commodityCodeType":"UK"
    }
}
```

Using own alcohol value

```
{
    "commodityCode":{
    "lwin":"112266220160600750",
    "commodityCodeType":"UK",
    "alcoholValue": "13"
    }
}
```

XML Request

Using Liv-ex alcohol value

```
<commodityCodeRequest>
    <commodityCode>
        <lwin>1637885200603000</lwin>
        <commodityCodeType>UK</commodityCodeType>
    </commodityCode>
</commodityCodeRequest>
```

Using own alcohol value

```
<commodityCodeRequest>
    <commodityCode>
        <lwin>1637885200603000</lwin>
        <commodityCodeType>UK</commodityCodeType>
        <alcoholValue>12</alcoholValue>
    </commodityCode>
</commodityCodeRequest>
```

Sample Response Body

The Commodity Code service will respond with HTTP Code 200 OK in a successful response.

Response parameters

Name	Description
lwin	LWIN16/18 **Type:** alphanumeric
commodityCode	Commodity code generated based on available data **Type:** alphanumeric
commodityCodeType	Specified in the response commodity code standard **Type:** alphanumeric

LWIN information.

Successful responses contain an "lwinStatus" element after the API information element ("apiInfo") and before the main body of the response. This states:

1. inputLwin – the LWIN codes requested
2. status – the status of the LWIN requested ('live' or 'combined')
3. combinedReference – if combined the actual live LWIN

JSON Response

The response is sent per request.

Alcohol value is available in the Liv-ex database for the specified LWIN

```
{
    "status": "OK",
    "httpCode": "200",
    "message": "Request completed successfully",
    "internalErrorCode": "R001",
    "apiInfo": {
        "version": "1.0",
        "timestamp": 1609768086331,
        "provider": "Liv-ex"
    },
    "lwinStatus": {
        "inputLwin": "1170126",
        "status": "live",
```

```
        "combineReference": null
    },
    "commodityCode": {
        "lwin": "117012620180600750",
        "commodityCode": "2204212790",
        "commodityCodeType": "UK"
    },
    "errors": null
}
```


Alcohol value is not available in the Liv-ex database for the specified LWIN


```
{
    "status": "OK",
    "httpCode": "200",
    "message": "Request completed successfully",
    "internalErrorCode": "R001",
    "apiInfo": {
        "version": "1.0",
        "timestamp": 1609768307379,
        "provider": "Liv-ex"
    },
    "lwinStatus": {
        "inputLwin": "1000131",
        "status": "combined",
        "combineReference": "1316384"
    },
    "commodityCode": {
        "lwin": "131638419750600750",
        "commodityCode": "220421",
        "commodityCodeType": "UK"
    },
    "errors": null
}
```


Alcohol value is provided in the API request


```
{
    "status": "OK",
    "httpCode": "200",
    "message": "Request completed successfully",
    "internalErrorCode": "R001",
    "apiInfo": {
        "version": "1.0",
        "timestamp": 1609769238659,
        "provider": "Liv-ex"
    },
    "lwinStatus": {
        "inputLwin": "1000131",
        "status": "combined",
        "combineReference": "1316384"
    },
    "commodityCode": {
        "lwin": "131638419750600750",
        "commodityCode": "2204211110",
        "commodityCodeType": "UK"
    },
    "errors": null
}
```


Invalid JSON response


```
{
    "status": "OK",
    "httpCode": "200",
    "message": "Request completed successfully",
    "internalErrorCode": "R001",
```

```
    "apiInfo": {
        "version": "1.0",
        "timestamp": 1609769279017,
        "provider": "Liv-ex"
    },
    "commodityCode": {
        "lwin": "100013119750600750",
        "commodityCodeType": "",
        "alcoholValue": "13"
    },
    "errors": {
        "error": [
            {
                "code": "V000",
                "message": "Mandatory field missing"
            }
        ]
    }
}
```


XML Response

The response is sent per request.


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<commodityCodeResponse>
    <Status>OK</Status>
    <HttpCode>200</HttpCode>
    <Message>Request completed successfully</Message>
    <InternalErrorCode>R001</InternalErrorCode>
    <ApiInfo>
        <Version>1.0</Version>
        <Timestamp>2021-01-04T14:14:23.576Z</Timestamp>
        <Provider>Liv-ex</Provider>
    </ApiInfo>
    <lwinStatus>
        <inputLwin>1637885</inputLwin>
        <status>live</status>
        <combineReference xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
    </lwinStatus>
    <commodityCode>
        <lwin>1637885200603000</lwin>
        <commodityCode>2204229310</commodityCode>
        <commodityCodeType>UK</commodityCodeType>
    </commodityCode>
    <errors xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
</commodityCodeResponse>
```


Invalid XML Response


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<commodityCodeResponse>
    <Status>OK</Status>
    <HttpCode>200</HttpCode>
    <Message>Request completed successfully</Message>
    <InternalErrorCode>R001</InternalErrorCode>
    <ApiInfo>
        <Version>1.0</Version>
        <Timestamp>2021-01-04T14:15:12.158Z</Timestamp>
        <Provider>Liv-ex</Provider>
    </ApiInfo>
```

```
    <commodityCode>
        <lwin>1637885200603000</lwin>
        <commodityCodeType>K</commodityCodeType>
        <alcoholValue>12</alcoholValue>
    </commodityCode>
    <errors>
        <error>
            <code>V161</code>
            <message>Invalid / incorrect commodity code type: K. Possible values are
'UK' or 'EU'.</message>
        </error>
    </errors>
</commodityCodeResponse>
```

6. Response Codes

This section describes the response codes that will be returned by the Exchange Integration services.

Code	Message
R000	Request was unsuccessful
R001	Request completed successfully
R002	Request partially completed

6.1 Request validation error codes

Code	Message
V000	("Mandatory field missing")
V006	("Invalid LWIN number.")
V160	("Commodity code cannot be generated. ")
V161	("Invalid / incorrect commodity code type: <value>. Possible values are 'UK', 'EU' or 'SG' .")

6.2 HTTP Status codes

HTTP defines a bunch of meaningful status codes that can be returned from our API. These can be leveraged to help our API Merchants/consumers route their responses accordingly:

Code	Message
200 OK	Response to a successful GET, POST, PUT, DELETE. Can also be used for a POST that doesn't result in a creation.
201 Created	Response to a POST that results in a creation.
202 Accepted	The request has been accepted and will be processed later. It is a classic answer to asynchronous calls (for better UX or performances).

204 No Content	Response to a successful request that won't be returning a body (like a DELETE request)
400 Bad Request	The request is malformed, such as if the body does not parse
401 Unauthorized	When no and/or invalid authentication details are provided. Can also be used to trigger an auth popup if API is used from a browser
403 Forbidden	When authentication succeeded but authenticated user doesn't have access to the resource
404 Not Found	When a non-existent resource is requested
405 Method Not Allowed	When an HTTP method is being requested that isn't allowed for the authenticated user
406 Not Acceptable	Nothing matches the Accept-* Header of the request. As an example, you ask for an XML formatted resource, but it is only available as JSON.
409 Conflict	Indicates one or more supplied parameters are triggering a validation error. A relevant TR code should be returned in the response.
410 Gone	Indicates that the resource at this end point is no longer available. Useful as a blanket response for old API versions
415 Unsupported Media Type	If incorrect content type was provided as part of the request
422 Unprocessable Entity	Used for validation errors. Should be used if the server cannot process the entity, e.g. if an image cannot be formatted or mandatory fields are missing in the payload.
429 Too Many Requests	When a request is rejected due to rate limiting
500 Internal Server Error	The general catch-all error when the server-side throws an exception. The request may be correct, but an execution problem has been encountered at our end.

7. Excluded codes (EU/UK standards)

Commodity code				
22	04	21	06	00
22	04	21	07	00
22	04	21	08	00
22	04	21	09	00
22	04	21	94	61
22	04	21	94	61
22	04	21	94	71
22	04	21	94	71
22	04	21	94	81
22	04	21	94	91
22	04	21	94	95
22	04	21	96	61
22	04	21	96	71
22	04	21	96	81
22	04	21	96	91
22	04	21	96	95
22	04	21	96	61
22	04	21	96	71
22	04	21	96	81
22	04	21	96	91
22	04	21	96	95
22	04	21	98	61
22	04	21	98	71
22	04	21	98	81
22	04	21	98	91
22	04	21	98	95
22	04	21	98	61
22	04	21	98	71
22	04	21	98	81

22	04	21	98	91
22	04	21	98	95
22	04	22	10	00
22	04	22	94	61
22	04	22	94	71
22	04	22	94	81
22	04	22	94	91
22	04	22	94	95
22	04	22	94	61
22	04	22	94	71
22	04	22	94	81
22	04	22	94	91
22	04	22	94	95
22	04	22	96	61
22	04	22	96	71
22	04	22	96	81
22	04	22	96	91
22	04	22	96	95
22	04	22	96	61
22	04	22	96	71
22	04	22	96	81
22	04	22	96	91
22	04	22	96	95
22	04	22	98	61
22	04	22	98	71
22	04	22	98	81
22	04	22	98	91
22	04	22	98	95
22	04	22	98	61
22	04	22	98	71
22	04	22	98	81
22	04	22	98	91

22	04	22	98	95
22	04	29	10	00
22	04	29	94	61
22	04	29	94	71
22	04	29	94	81
22	04	29	94	91
22	04	29	94	95
22	04	29	94	61
22	04	29	94	71
22	04	29	94	81
22	04	29	94	91
22	04	29	94	95
22	04	29	96	61
22	04	29	96	71
22	04	29	96	81
22	04	29	96	91
22	04	29	96	95
22	04	29	96	61
22	04	29	96	71
22	04	29	96	81
22	04	29	96	91
22	04	29	96	95
22	04	29	98	61
22	04	29	98	71
22	04	29	98	81
22	04	29	98	91
22	04	29	98	95
22	04	29	98	61
22	04	29	98	71
22	04	29	98	81
22	04	29	98	91
22	04	29	98	95

22	04	30		
22	08	40	11	00
22	08	40	31	00
22	08	40	39	00
22	08	40	51	00
22	08	40	91	00
22	08	40	99	00
22	08	90	11	00
22	08	90	19	00
22	08	90	33	00
22	08	90	38	00
22	08	90	41	00
22	08	90	71	00
22	08	90	91	00
22	08	90	99	00