LIV EX

THE FINE WINE MARKET

Offex Data v1

Document Revision 1.0
Date of Issue: 10/12/25
Date of revision: 10/12/25

Barnabas Mullan-Gage
Senior Business Analyst

Table of Contents

1. Purpose

This document is to provide the API end point information and operational notes for the Offex Data API.

2. Glossary of Terms

Term	Meaning
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.
Last Offex	The last occurring Offex transaction matching the LWIN11 provided.
My Last Offex	The last submitted Offex transaction by your company, matching the LWIN11 provided.
High	The highest Offex value found in the specified timeframe matching the LWIN11 provided
Low	The lowest Offex value found in the specified timeframe matching the LWIN11 provided
Average	The calculated average Offex value found in the specified timeframe matching the LWIN11 provided. Calculations consider quantity of cases.

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 Offex Data service (POST method)

Description

This service allows users to request information about Offex data for specified products, either by LWIN codes or passing a listID (generated by using the Custom Lists family of APIs). The API returns value, date and quantity information about the Offex transactions.

Base URI

/data/v1/offexData

URI Pagination Parameters

Name	Mandatory	Description
?limit	N default = 1000	Values: Any value between '1' and '10000' **Type:** integer
?offset	N default = 1	**Type:** alphanumeric

Example base URI including pagination: /data/v1/offexData?offset=1&limit=25

Request Parameters

Name	Mandatory	Description
lwin	Y (unless listId is supplied)	A list of valid LWIN11, LWIN16 or LWIN18s. LWINs submitted must be of the same length. Type: array. Max 1500 LWINs
listID	Y (unless LWIN is supplied)	When submitting a listID, LWIN is no longer a mandatory parameter. All LWINs submitted to the listID provided will be called against. Values: listGUID or Stock list

		Type: alphanumeric This parameter should be ignored when “lwin” value(s) provided in the request.
lwinDepth	N (default = LWIN11)	For use when submitting a listID, to control how much of the LWIN codes stored in the list to use. Possible values are LWIN11, LWIN16 and LWIN18. This parameter should be ignored if LWIN values are provided in the request Type: Alphanumeric
currency	Y	Any possible LX currency, including the btt option. Values: ‘gbp’, ‘eur’, ‘chf’, ‘usd’, ‘hkd’, ‘jpy’, ‘sgd’, ‘gbp/btt’, ‘eur/btt’, ‘chf/btt’, ‘usd/btt’, ‘hkd/btt’, ‘jpy/btt’, ‘sgd/btt’ Type: alphanumeric Not case sensitive
includeLastOffex	N (Default false)	When true, include the last Offex results cluster in the response Type: boolean
includeMyLastOffex	N (Default false)	When true, include the last merchant submitted Offex results cluster in the response Type: Boolean
includeAverageOffex	N (Default false)	When true, include the average Offex results cluster in the response Type: Boolean

includeHighLowOffex	N (Default false)	When true, include the High and Low Offex results cluster in the response Type: boolean

Sample Request Body

JSON Request

```
{
    "offexData":{
        "listID":"94b71bab-8ca4-4e64-bdc8-abb8becf8512",
        "lwin": [],
        "lwinDepth":"lwin18",
        "currency":"usd",
        "includeLastOffex": true,
        "includeMyLastOffex": true,
        "includeAverageOffex": true,
        "includeHighLowOffex": false,
    }
}
```

XML Request

```
<root>
    <offexData>
        <listID>94b71bab-8ca4-4e64-bdc8-abb8becf8512</listID>
        <lwinDepth>lwin18</lwinDepth>
        <currency>usd</currency>
        <includeLastOffex>true</includeLastOffex>
    </offexData>
</root>
```

Response Parameters

Name	Description
currency	The currency format of the response Type: Alphanumeric
listID	The listID as specified in the request. When LWIN was used in the request, returns as null. Type: 128-bit hexadecimal
lineID	Unique identifier of the line within the list specified. When LWIN was used in the request, returns as null. Type: 128-bit hexadecimal
yourProductID	The defined product code attributed for this line within the list specified. When LWIN was used in the request, returns as null. Type: alphanumeric

lwin	The LWIN code of the product returned. This can be LWIN11/16/18 length. Type: alphanumeric
lwinName	The LWIN vintage specific display name description e.g. 'Chateau Grand-Puy-Lacoste 5eme Cru Classe, Pauillac' Type: alphanumeric
vintage	The vintage of the wine corresponding to the data returned. Type: alphanumeric
packSize	The pack size corresponding to the data returned. Type: alphanumeric Example: 12
bottleSize	The bottle size corresponding to the data returned in ml (millilitres) Type: alphanumeric Example: 00750
value	Value as converted to a 9L case (12x75) equivalent Type: Double
date	Type: Date (epoch if JSON)
qty	Quantity Type: Integer
highOffex	Type: Double
lowOffex	Type: Double
hlaTimeframe	Defines the timeframe applied to calculate the high, low and average Offex. Type: Alphanumeric

Sorting Order

a. When LWINs are provided in the request: present LWINs in the same order as in the request

b. When valid listID is provided in the request: present LWINs in the order they are recorded on the list (by Line ID creation date oldest to newest)

Sample Response Body

JSON Response

```
{
    "status": "OK",
    "httpCode": "200",
    "message": "Request completed successfully",
    "internalErrorCode": "R001",
    "apiInfo": {
        "version": "1.0",
        "timestamp": 1744377706315,
        "provider": "Liv-ex"
    },
    "pageInfo": {
        "totalResults": 1,
        "limit": 10000,
        "offset": 1
    },
    "offexDataResponse": {
        "currency": "gbp",
        "listID": "94b71bab-8ca4-4e64-bdc8-abb8becf8512",
        "data": [
            {
                "lineID": "94b71bab-8ca4-4e64-bdc8-abb8becf8512",
                "yourProductID": null,
                "hlaTimeframe": "30 days",
                "lwin": "147895820180600750",
                "lwinName": "Domaine Louis Moreau, Chablis Grand Cru, Vaudesir",
                "vintage": "2018",
                "packSize": "06",
                "bottleSize": "00750",
                "offexData": {
                    "value": "1000",
                    "date": "1744227590957",
                    "qty": "5"
                },
                "myLastOffexData": {
                    "value": "1500",
                    "date": "1744227590957",
                    "qty": "20"
                },
                "highLowOffexData": {
                    "highOffex": "1500",
                    "lowOffex": "1000"
                },
                "averageOffexData": {
                    "value": "1400",
                    "qty": "20.5"
                }
            }
        ]
    }
}
```

XML Response

```
<?xml version="1.0" encoding="UTF-8" ?>
<root>
<status>OK</status>
<httpCode>200</httpCode>
<message>Request completed successfully</message>
<internalErrorCode>R001</internalErrorCode>
```

```
<apiInfo>
<version>1.0</version>
<timestamp>1744377706315</timestamp>
<provider>Liv-ex</provider>
</apiInfo>
<pageInfo>
<totalResults>1</totalResults>
<limit>10000</limit>
<offset>1</offset>
</pageInfo>
<offexDataResponse>
<currency>gbp</currency>
<listID>94b71bab-8ca4-4e64-bdc8-abb8becf8512</listID>
<data>
<lineID>94b71bab-8ca4-4e64-bdc8-abb8becf8512</lineID>
<yourProductID />
<hlaTimeframe>30 days</hlaTimeframe>
<lwin>147895820180600750</lwin>
<lwinName>Domaine Louis Moreau, Chablis Grand Cru, Vaudesir</lwinName>
<vintage>2018</vintage>
<packSize>06</packSize>
<bottleSize>00750</bottleSize>
<offexData>
<value>1000</value>
<date>1744227590957</date>
<qty>5</qty>
</offexData>
<myLastOffexData>
<value>1000</value>
<date>1744227590957</date>
<qty>5</qty>
</myLastOffexData>
<highLowOffexData>
<highOffex>1500</highOffex>
<lowOffex>1000</lowOffex>
</highLowOffexData>
<averageOffexData>
<value>1000</value>
<qty>5</qty>
</averageOffexData>
</data>
</offexDataResponse>
</root>
```

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	Trigger
V018	Mandatory field missing(%s).	One of, no LWIN and no ListID provided, or no Currency provided.
V039	You do not have permission to access (%). Please contact Liv-ex.	The data point requested is one you do not have access to. Speak to your account manager to gain access to new data points.
V058	Invalid / incorrect LWIN: [${v}]. Please provide a valid LWIN11, LWIN16 or LWIN18 code.	When all the supplied LWIN codes in the LWIN array are invalid.
V061	Invalid / incorrect currency: []. Possible values are 'gbp', 'eur', 'chf', 'usd', 'hkd', 'jpy', 'sgd', 'gbp/btt', 'eur/btt', 'chf/btt', 'gbpbtt', 'eurbtt', 'chfbtt'."	Invalid characters passed in request parameter “currency”.
V125	Requests with multiple LWINs must contain LWINs of the same length.	When LWINs of varying lengths are supplied in the LWIN array
V174	Invalid / incorrect listID: [%]. Please provide a valid listID value.	The listID specified does not belong to you, or has been deleted.

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.