LIV | EX

THE FINE WINE MARKET

Market Price and Benchmark API v1

Document Revision 1.0
Date of Issue: 21/09/2024
Date of revision: 21/09/2024

Barnabas Mullan
Senior Business Analyst

Table of Contents

1. **Purpose** 3
2. **Glossary of Terms** 3
3. **Technical Standards** 3
4. **Request Header** 4
5. **API Listing** 5
   5.1 Market Price and Benchmark API Service (POST Method) 5
6. **Response Codes** 12
   6.1 Request validation error codes 13
   6.2 HTTP Status codes 14

1. Purpose

To provide the API end point information and examples of the web services available for the Market Price and Benchmark 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.
lineID	Identifier for each individual product list uploaded to the Liv-ex system, created via the List Manager POST API
listID	Identifier for each individual line within the list, created via the Line Manager POST API
Market Price	The Market Price is calculated based on publicly advertised prices sourced from leading merchants in the EU and Switzerland. It provides a guide as to the price you are likely to pay for SIB-compliant stock in the market

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 contents type in the request header. If the user does not provide any information, then the default content type will be JSON.
- The project will support ISO 8601.
- The project will only support HTTPS protocol for client and server communications.
- The API will support the following methods:
  - POST for create operation
  - PATCH for edit operation
  - DELETE for delete operation
- Pretty printing for output readability only is supported if required.
- Compression for bandwidth savings are used.
- For HTTP users who can only work on GET & POST methods, we provide a Header 'X-HTTP-Method-Override' for DELETE

- Authentication mechanism will be custom based on CLIENT_KEY and CLIENT_SECRET.
- For any PUSH services we require a direct POST URL which should be backed by as service capable of accepting and process XML payload as POST request.
- 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.

Name	Mandatory	Description
CLIENT_KEY	Y	A valid user GUID which will be unique for each merchant
CLIENT_SECRET	Y	Password/Secret for the user CLIENT_KEY
ACCEPT	Y	Accept header is a way for the user 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	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 Headers

```
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 Market Price and Benchmark API Service (POST Method)

Description

This service allows users to request information on various Liv-ex pricing datapoints using an LWIN array or a ListID. The API service returns information about the price, date and confidence of specified products, and can optionally apply a bottle size premium when requested by the user.

Base URI

/data/v1/marketPriceAndBenchmark

Request Parameters

Name	Mandatory	Description
lwin	Y (Unless listID is supplied)	A list of valid lwin18, lwin16 or lwin11. LWINs submitted must be of the same length. Type: Alphanumeric Array, max 1500 values
listID	Y (Unless LWIN is supplied)	When submitting a listID, LWIN is no longer a mandatory parameter, and will call results for all LWINs stored within the specified list. This parameter will be ignored if the "lwin" values are provided in the request. Accepts both "Stock List" and custom list IDs.

		Type: 128-bit hexadecimal
lwinDepth	N (Default LWIN11)	Determines the length of LWIN considered when using the listID request parameter. This parameter will be ignored if the "lwin" values are provided in the request. Values: lwin11, lwin16 and lwin18. 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
priceDate	N	Used to specify a historic date for which data will be recalled. This can be any date in the past but the range of data available will be restricted and controlled by your historic data licence. If unspecified, the prices returned will be the most recent possible values. Type: alphanumeric (yyyy-mm-dd)
includeLastTrade	N	Indicate whether you would like to receive data on the last trade price in the API response. Type: Boolean true/false

includeMarketPrice	N	Indicate whether you would like to receive data on the market price value and date in the API response. Type: Boolean true/false
includeMarketPriceConfidence	N	Indicate whether you would like to receive data on the market price confidence in the API response. marketPriceConfidence as a response parameter will return a traffic light indicator based on the marketPriceConfidenceScore threshold. Type: Boolean true/false
includeValuation	N	Indicate whether you would like to receive data on the valuation value and date in the API response. Type: Boolean true/false
includeValuationConfidence	N	Indicate whether you would like to receive data on the valuation confidence in the API response. valuationPriceConfidence as a response parameter will return a traffic light indicator based on the valuationPriceConfidenceScore threshold. Type: Boolean true/false
includeBottlePremium	N	Determines whether returned values have a bottle size premium automatically applied. Requires a minimum lwin16 depth. Type: Boolean true/false

benchmarkPrices	N	Possible values: “US”, “HK”, “SG”, “Negociant” Type: alphanumeric array Not case sensitive.

Operational Notes:

Price data values are returned differently depending on the LWIN length submitted.

LWIN Length	Data Response
LWIN11	Considers all pack and bottle options for the vintage specified by the LWIN11 and returns a price normalised to 9 litres.
LWIN16	Considers prices that correspond to the bottle size format specified by the LWIN16.
LWIN18	Considers prices that correspond to the bottle size and pack size specified by the LWIN18.

Request examples

Sample JSON Request

```
{
    "marketPriceAndBenchmark": {
        "listID": "94b71bab-8ca4-4e64-bdc8-abb8becf8512",
        "currency": "usd",
        "lwinDepth": "lwin18",
        "includeBottlePremium": true,
        "includeMarketPrice": true,
        "includeMarketPriceConfidence": true,
        "includeLastTrade": true,
        "includeValuation": true,
        "includeValuationConfidence": true,
        "benchmarkPrices": [
            "US",
            "HK",
            "SG",
            "negociant"
        ]
    }
}
```

Sample XML Request

```
<root>
    <marketPriceAndBenchmark>
        <currency>gbp</currency>
        <lwin>10110922010</lwin>
        <includeBottlePremium>false</includeBottlePremium>
        <includeLastTrade>true</includeLastTrade>
        <includeMarketPrice>true</includeMarketPrice>
        <benchmarkPrices>US</benchmarkPrices>
        <benchmarkPrices>SG</benchmarkPrices>
        <benchmarkPrices>HK</benchmarkPrices>
        <benchmarkPrices>Negociant</benchmarkPrices>
    </marketPriceAndBenchmark>
</root>
```

Response Parameters

Name	Description
currency	The currency format of the price data contained the response. Type: Alphanumeric
listID	listID is 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.
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
lastTradeValue	Type: double
lastTradeDate	Type: Date (epoch if JSON)
marketPriceValue	Type: double
marketPriceDate	Type: Date (epoch if JSON)
marketPriceConfidence	Traffic light indication of the Market Price confidence Type: Alphanumeric
marketPriceConfidenceScore	Always null. Ignore this response parameter.

valuationPriceValue	Type: double
valuationPriceDate	Type: Date (epoch if JSON)
valuationPriceConfidence	Traffic light indication of the Valuation confidence Type: Alphanumeric
valuationPriceConfidenceScore	Always null. Ignore this response parameter.
benchmarkType	Returns detail on which benchmark value has been returned in this cluster. Type: Alphanumeric
benchmarkValue	Type: double
benchmarkDate	Type: Date (epoch if JSON)
bottlePremiumIncluded	Confirms whether a bottle premium has been included in the calculation of this price value. Type: Boolean true/false

Response Examples

Sample JSON Response

```
{
    "status": "OK",
    "httpCode": "200",
    "message": "Request completed successfully",
    "internalErrorCode": "R001",
    "apiInfo": {
        "version": "1.0",
        "timestamp": 1728920464441,
        "provider": "Liv-ex"
    },
    "pageInfo": {
        "totalResults": 10068,
        "limit": 1,
        "offset": 1
    },
    "marketPriceAndBenchmarkResponse": {
        "currency": "usd",
        "listID": "94b71bab-8ca4-4e64-bdc8-abb8becf8512",
        "data": [
            {
                "lineID": "f36feb8b-e055-43ac-86a5-9426775a4d5e",
                "bottlePremiumIncluded": true,
                "yourProductID": "CBatailley2009",
                "lwin": "100632220090600750",
                "lwinName": "Chateau Batailley 5eme Cru Classe,
Pauillac",
                "vintage": "2009",
                "packSize": "06",
                "bottleSize": "00750",
                "valuationData": {
                    "marketPrice": {
                        "marketPriceValue": 352.89,
                        "marketPriceDate": 1728864000000,
                        "marketPriceConfidence": "Green",
                        "marketPriceConfidenceScore": null
                    },
                    "valuation": {
                        "valuationPriceValue": 352.89,
                        "valuationPriceDate": 1728604800000,
                        "valuationPriceConfidence": "Green",
                        "valuationPriceConfidenceScore": null
                    },
                    "lastTrade": {
                        "lastTradeValue": 281.20,
                        "lastTradeDate": 1522925099000
                    },
                    "benchmarkPrices": [
                        {
                            "benchmarkType": "US",
                            "benchmarkValue": 386.87,
                            "benchmarkDate": 1728259200000
                        },
                        {
                            "benchmarkType": "HK",
                            "benchmarkValue": 382.95,
                            "benchmarkDate": 1726185600000
                        },
                        {
                            "benchmarkType": "SG",
                            "benchmarkValue": 619.52,
                            "benchmarkDate": 1727654400000
                        },
                        {
                            "benchmarkType": "negociant",
                            "benchmarkValue": null,
                            "benchmarkDate": null
                        }
                    ]
                }
            }
        ]
    }
}
```

Invalid JSON Response

```
{
    "status": "Internal Server Error",
    "httpCode": "500",
    "message": "Request was unsuccessful",
    "internalErrorCode": "R000",
    "apiInfo": {
        "version": "1.0",
        "timestamp": 1728920614756,
        "provider": "Liv-ex"
    }
}
```

Sample XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<root>
    <Status>OK</Status>
    <HttpCode>200</HttpCode>
    <Message>Request completed successfully</Message>
    <InternalErrorCode>R001</InternalErrorCode>
    <ApiInfo>
        <Version>1.0</Version>
        <Timestamp>2024-10-14T15:45:16.681Z</Timestamp>
        <Provider>Liv-ex</Provider>
    </ApiInfo>
    <pageInfo>
        <totalResults>269</totalResults>
        <limit>1</limit>
        <offset>1</offset>
    </pageInfo>
    <marketPriceAndBenchmarkResponse>
        <currency>gbp</currency>
        <listID>stock list</listID>
        <data>
            <lineID>713c7df0-10b2-40b2-859d-1cb80034b827</lineID>
            <bottlePremiumIncluded>true</bottlePremiumIncluded>
            <lwin>113479120210600750</lwin>
            <lwinName>Domaine Vincent Paris, Cornas, La
Geynale</lwinName>
            <vintage>2021</vintage>
            <packSize>06</packSize>
            <bottleSize>00750</bottleSize>
            <valuationData>
                <marketPrice>
                    <marketPriceValue>160</marketPriceValue>
                    <marketPriceDate>
                    </marketPriceDate>
                </marketPrice>
                <lastTrade/>
                <benchmarkPrices>
                    <benchmarkType>US</benchmarkType>
                    <benchmarkValue>270</benchmarkValue>
                    <benchmarkDate>
                    </benchmarkDate>
                </benchmarkPrices>
                <benchmarkPrices>
                    <benchmarkType>SG</benchmarkType>
                </benchmarkPrices>
                <benchmarkPrices>
                    <benchmarkType>HK</benchmarkType>
                </benchmarkPrices>
                <benchmarkPrices>
                    <benchmarkType>negociant</benchmarkType>
                </benchmarkPrices>
            </valuationData>
        </data>
    </marketPriceAndBenchmarkResponse>
</root>
```

Invalid XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Response>
    <Status>Internal Server Error</Status>
    <HttpCode>500</HttpCode>
    <Message>Request was unsuccessful</Message>
    <InternalErrorCode>R000</InternalErrorCode>
    <ApiInfo>
        <Version>1.0</Version>
        <Timestamp>2024-10-14T16:45:36.830+01:00</Timestamp>
        <Provider>Liv-ex</Provider>
    </ApiInfo>
</Response>
```

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).	No LWIN and no ListID provided, or no Currency provided.
V039	You do not have permission to access (%). Please contact Liv-ex.	When requesting a datapoint you don't have access to.
V040	Requested date should be a valid past date in 'yyyy-MM-dd' format	When priceDate request parameter is provided and is invalid.
V041	Requested date should be a valid date within the last () years(s).	When priceDate request parameter value exceeds the historical license of the calling MID.
V058	Invalid / incorrect LWIN: [${v}]. Please provide a valid LWIN11, LWIN16 or LWIN18 code.	When all supplied LWIN codes in the LWIN array are invalid.
V061	Invalid / incorrect currency: [<value>]. 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.	When listID that does not belong to the merchant, is invalid, or no longer live, is provided in the listID parameter.
V175	The listID provided does not contain any LWINs of the specified lwinDepth [%].	When the lwinDepth parameter specifies a length of LWIN that is is not present in the specified listID. I.e. LWIN18 depth specified but the ListID only contains LWIN11s.

V190	Minimum lwinDepth to include Bottle Premiums is LWIN16.	When the request parameter inludeBottlePremium is True, and lwinDepth parameter value is LWIN11

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.