LIV|EX

THE FINE WINE MARKET

Time Series Export API v1

Document Revision 1.0
Date of Issue: 02 September 2021
Date of revision: 02 September 2021

Barnabas Mullan

Business Analyst

Table of Contents

1. Purpose

To provide the API end point information and examples of the web services available for the Time Series Export API.

2. Glossary of Terms

Term	Meaning
Liv-ex Index	Liv-ex has created several indices to track the price of fine wine, including the industry benchmark Liv-ex 100. The Liv-ex indices which track the prices of the most traded fine wines on the market, using the Liv-ex Mid Price.

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 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
- 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.

Param

Name	Mandatory	Description
CLIENT_KEY	Y	A valid merchant GUID which will be unique for each merchant.
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.

e.g.
CLIENT_KEY: 94B5CC70-BC3D-49C3-B636-C3C7552E543D
CLIENT_SECRET: merchantpasswd
ACCEPT: application/json
CONTENT-TYPE: application/json

Invalid header JSON response

```
{
    "status": "Unauthorized",
    "httpCode": "401",
    "message": "Request was unsuccessful",
    "livexCode": "R000"
    "apiInfo": {
        "version": "1.0",
        "timestamp": 1518524979121,
        "provider": "Liv-ex"
    }
}
```

Invalid header XML response

```
<Response>
    <Status>Unauthorized</Status>
    <HttpCode>401</Code>
    <Message>Request was unsuccessful.</Message>
    <LivexCode>R001</LivexCode>
     <ApiInfo>
        <Version>1.0</Version>
        <Timestamp>2021-07-01T11:12:30</Timestamp>
        <Provider>Liv-ex</Provider>
    </ApiInfo>
<Response>
```

5. API Listing

5.1 Time Series Export service (POST method)

Description

This service will be used to request market price information for a given LWIN11 code or index close values for given internal index. On receipt of one or more valid LWIN11 codes or valid index names, the service will return a series of data values for the LWIN11 codes and specified indexes for the time range specified. Any LWIN11 dates requested that fall outside your licensed limit for historical data will not be returned.

Base URI

data/v1/timeSeriesExport

Request Parameters

Name	Mandatory	Description
lwin	N	An array of one or more valid LWIN11 codes (unlimited). Type: 11-digit integer array Example: 10118721995
internalIndex	N	Name of the Liv-ex Index Not case sensitive Type: alphanumeric array
timeframe	N Default = "6month"	Period of time going back from today's date. Timeframe options of the data series in months or years. The possible values are "6month", "1year", "2year", "5year", "ytd", "maximum". *If value is invalid the service will default to "timeframe" = "6month".* *If both timeframe and dateFrom/dateTo parameters were specified then dateFrom/dateTo will be ignored.* Not case sensitive Type: alphanumeric
dateFrom	N	Start date of data series. *If value is invalid the service will default to "timeframe" = "6month".* A valid date in yyyy-mm-dd format e.g. 2018-07- 30. Type: alphanumeric, ISO8601 format
dateTo	N	End date of data series. *If value is invalid the service will default to "timeframe" = "6month ".* A valid date in yyyy-mm-dd format e.g. 2018-07- 30. Type: alphanumeric, ISO8601 format
currency	N Default = "GBP"	Specify the currency of pricing data requested. Possible values are "GBP", "USD", "JPY", "EUR", "CHF", "SGD", "HKD", "GBP/btt", "EUR/btt", "CHF/btt". Not case sensitive *Not relevant to internalIndex series* Type: alphanumeric

Sample Request Body

JSON Request

```
{
    "timeSeriesExport": {
        "timeframe": "1year",
        "dateFrom":"",
        "dateTo":"",
        "lwin": ["12294592019"],
        "internalIndex": ["Liv-ex Fine Wine 50"],
        "currency": "GBP"
    }
}
```

XML Request

```
<timeSeriesExportRequest>
<timeSeriesExport>
    <timeframe>1year</timeframe>
    <lwin>12294592019</lwin>
    <internalIndex>Liv-ex Fine Wine 50</internalIndex>
    <currency>GBP</currency>
    </timeSeriesExport>
</timeSeriesExportRequest>
```

Sample Response Body

Response parameters

Name	Description
licencedHistoricalDepth	The number of years of historical data the API caller is licensed for. Type: integer
timeframe	Type: alphanumeric
dateFrom	Start date of data series. Type: alphanumeric, ISO8601 format Example (JSON): 1549537950898 Example (XML): 2019-02-07T11:12:30
dateTo	End date of data series. Type: alphanumeric, ISO8601 format Example (JSON): 1549537950898 Example (XML): 2019-07-07T11:12:30
currency	Type: alphanumeric
group	Classification of data in response. Possible values are "lwin" or " internalIndex" Type: alphanumeric
lwin	LWIN11 Null when group = " internalIndex " Type: alphanumeric
lwinName	Vintage-specific LWIN wine name description e.g. 'Chateau Grand-Puy-Lacoste 5eme Cru Classe, Pauillac' Null when group = " internalIndex " Type: alphanumeric
vintage	Vintage Null when group = " internalIndex " Type: alphanumeric
indexName	Name of the internal index Null when group = "lwin" Type: alphanumeric
indexShortName	Short name of the Liv-ex index, e.g. LX50 Type: alphanumeric
components	Type: alphanumeric, array of LWIN11 Always null for groups "lwin" and " internalIndex"
date	Date of specific data point Type: alphanumeric, ISO 8601 Epoch time if JSON Example (JSON): 1549537950898 Example (XML): 2019-02-07T11:12:30
value	When group = lwin -> API returns market prices When group = internalIndex -> index values Type: double

JSON Response

The response is sent per request. The example below shows the response for a request with one LWIN 11. The response example has been shortened to 3 response values.

A more complex response payload featuring multiple LWINs and indices is shown in Appendix 1.


```
{
    "status": "OK",
    "httpCode": "200",
    "message": "Request completed successfully",
    "internalErrorCode": "R001",
    "apiInfo": {
        "version": "1.0",
        "timestamp": 1627652904200,
        "provider": "Liv-ex"
    },
    "timeSeriesExport": {
        "timeframe": "6month",
        "dateFrom": null,
        "dateTo": null,
        "currency": "GBP",
        "groups": [
            {
                "group": "lwin",
                "list": [
                    {
                        "lwin": "13194602017",
                        "lwinName": "Giuseppe Rinaldi, Barolo, Tre Tine",
                        "vintage": "2017",
                        "indexName": null,
                        "indexShortName": null,
                        "components": null,
                        "series": {
                            "values": [
                                {
                                    "date": 1623801600000,
                                    "value": 0000.0
                                },
                                {
                                    "date": 1623715200000,
                                    "value": 0000.0
                                },
                                {
                                    "date": 1623628800000,
                                    "value": 0000.0
                                }
                            ]
                        }
                    }
                ]
            }
        ]
    },
    "errors": null
}
```


Invalid JSON response


```
{
    "status": "Bad Request",
    "httpCode": "400",
    "message": "Request was unsuccessful",
    "internalErrorCode": "R000",
    "apiInfo": {
        "version": "1.0",
        "timestamp": 1630659569006,
        "provider": "Liv-ex"
    },
    "timeSeriesExport": null,
    "errors": {
```

```
        "error": [
            {
                "code": "V006",
                "message": "Invalid LWIN number."
            }
        ]
    }
}
```

XML Response

The response is sent per request. The example below shows the response for a request with one LWIN 11. The response example has been shortened to 3 response values.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<timeSeriesExportResponse>
    <Status>OK</Status>
    <HttpCode>200</HttpCode>
    <Message>Request completed successfully</Message>
    <InternalErrorCode>R001</InternalErrorCode>
    <ApiInfo>
        <Version>1.0</Version>
        <Timestamp>2021-08-06T10:45:04.219Z</Timestamp>
        <Provider>Liv-ex</Provider>
    </ApiInfo>
    <timeSeriesExport>
        <timeframe>1year</timeframe>
        <dateFrom xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
        <dateTo xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
        <currency>GBP</currency>
        <groups>
            <group>lwin</group>
            <list>
                <lwin>12294592019</lwin>
                <lwinName>Marchand-Tawse, Corton Grand Cru, Rouge</lwinName>
                <vintage>2019</vintage>
                <indexName xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
                <indexShortName xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
                <components xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
                <series>
                    <values>
                        <date>2020-12-18</date>
                        <value>0000.0</value>
                    </values>
                    <values>
                        <date>2020-12-25</date>
                        <value>0000.0</value>
                    </values>
                    <values>
                        <date>2021-01-01</date>
                        <value>0000.0</value>
                    </values>
                </series>
            </list>
        </groups>
```

Invalid XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<timeSeriesExportResponse>
    <Status>Bad Request</Status>
    <HttpCode>400</HttpCode>
    <Message>Request was unsuccessful</Message>
    <InternalErrorCode>R000</InternalErrorCode>
    <ApiInfo>
        <Version>1.0</Version>
        <Timestamp>2021-09-03T08:58:50.380Z</Timestamp>
        <Provider>Liv-ex</Provider>
    </ApiInfo>
    <timeSeriesExport xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
    <errors>
        <error>
            <code>V006</code>
            <message>Invalid LWIN number.</message>
        </error>
    </errors>
</timeSeriesExportResponse>
```

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
V006	Invalid LWIN number
V125	Requests with multiple LWINs must contain LWINs of the same length.
V169	Invalid / incorrect internalIndex name: [%s].
V172	You do not have permission to access historic values. Please contact your account manager.

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. Appendix 1 – Longer API response

The JSON response below shows a full response for a single LWIN with the timeframe of 1 year.


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

```
        "version": "1.0",
        "timestamp": 1628245789606,
        "provider": "Liv-ex"
    },
    "timeSeriesExport": {
        "timeframe": "1year",
        "dateFrom": null,
        "dateTo": null,
        "currency": "GBP",
        "groups": [
            {
                "group": "lwin",
                "list": [
                    {
                        "lwin": "12294592019",
                        "lwinName": "Marchand-Tawse, Corton Grand Cru, Rouge",
                        "vintage": "2019",
                        "indexName": null,
                        "indexShortName": null,
                        "components": null,
                        "series": {
                            "values": [
                                {
                                    "date": 1608249600000,
                                    "value": 100.0
                                },
                                {
                                    "date": 1608854400000,
                                    "value": 100.0
                                },
                                {
                                    "date": 1609459200000,
                                    "value": 100.0
                                },
                                {
                                    "date": 1610064000000,
                                    "value": 100.0
                                },
                                {
                                    "date": 1610668800000,
                                    "value": 100.0
                                },
                                {
                                    "date": 1611273600000,
                                    "value": 100.0
                                },
                                {
                                    "date": 1611878400000,
                                    "value": 100.0
                                },
                                {
                                    "date": 1612483200000,
                                    "value": 100.0
                                },
                                {
                                    "date": 1613088000000,
                                    "value": 100.0
                                },
                                {
                                    "date": 1613692800000,
                                    "value": 100.0
                                },
                                {
                                    "date": 1614297600000,
                                    "value": 100.0
                                },
```

```
                            {
                                "date": 1614902400000,
                                "value": 100.0
                            },
                            {
                                "date": 1615507200000,
                                "value": 100.0
                            },
                            {
                                "date": 1616112000000,
                                "value": 100.0
                            },
                            {
                                "date": 1616716800000,
                                "value": 100.0
                            },
                            {
                                "date": 1617321600000,
                                "value": 100.0
                            },
                            {
                                "date": 1617926400000,
                                "value": 100.0
                            },
                            {
                                "date": 1618531200000,
                                "value": 100.0
                            },
                            {
                                "date": 1619136000000,
                                "value": 100.0
                            },
                            {
                                "date": 1619740800000,
                                "value": 100.0
                            },
                            {
                                "date": 1620345600000,
                                "value": 100.0
                            },
                            {
                                "date": 1620950400000,
                                "value": 100.0
                            },
                            {
                                "date": 1621555200000,
                                "value": 100.0
                            },
                            {
                                "date": 1622160000000,
                                "value": 100.0
                            },
                            {
                                "date": 1622764800000,
                                "value": 100.0
                            },
                            {
                                "date": 1623369600000,
                                "value": 100.0
                            }
                        ]
                    }
                }
            ]
        }
    ]
```

```
    },
    "errors": null
}
```