LIV|EX

THE FINE WINE MARKET

Chart Data Performance API v1

Document Revision 1.0
Date of Issue: 17 August 2021
Date of revision: 17 August 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 Chart Data Performance 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 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
- 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 Chart Data Performance service (POST method)

Description

This service will be used to request index and market price performance data.

Base URI

data/v1/chartDataPerformance

Request Parameters

Name	Mandatory	Description
lwin	N	An array of one or more valid LWIN11 codes. There is not an implemented limit on the number of LWIN11 codes you can request in the array, however requesting a large number of LWIN11 codes will result in a slower API response. Type: 11-digit integer array Example: 10118721995
internalIndex	N	Liv-ex index (for comparison of the wine against an index e.g. with LVX50, LVX 100 etc) Type: alphanumeric array Example: "Liv-ex Fine Wine 100", "Liv-ex Bordeaux 500". A full list of valid index names can be requested using Indices List v1 GET service.
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 relevant to the internalIndex series.* Not case sensitive Type: alphanumeric

Sample Request Body

JSON Request

```
{
    "chartDataPerformance": {
        "currency": "",
        "lwin": [10140332009,10127812011],
        "internalIndex":["Liv-ex Fine Wine 50","Liv-ex Fine Wine 100"]
    }
}
```

XML Request

```
<chartDataPerformanceRequest>
<chartDataPerformance>
<currency>gbp</currency>
<lwin>10127812009</lwin>
<lwin>10127812010</lwin>
<internalIndex>Liv-ex Fine Wine 50</internalIndex>
<internalIndex>Liv-ex Fine Wine 100</internalIndex>
</chartDataPerformance>
</chartDataPerformanceRequest>
```

Sample Response Body

Response parameters

Name	Description
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
value	For the "lwin" group the API will return market price For the " internalIndex " group the API will return index values Type: double
monthOnMonthChange	The change in value in 1 month from the date of the API call. Type: double
yearToDateChange	The change in value since the beginning of the current year to the date of the API call. Type: double
oneYearChange	The change in value in 1 year from the date of the API call. Type: double
twoYearChange	The change in value in 2 years from the date of the API call. Type: double
fiveYearChange	The change in value in 5 years from the date of the API call. Type: double

JSON Response

The response is sent per request. The example below shows the response for the JSON Request example.


```
{
    "status": "OK",
    "httpCode": "200",
    "message": "Request completed successfully",
    "internalErrorCode": "R001",
    "apiInfo": {
        "version": "1.0",
        "timestamp": 1629195516225,
        "provider": "Liv-ex"
    },
    "chartDataPerformance": {
        "currency": "GBP",
        "groups": [
            {
                "group": "lwin",
                "list": [
                    {
                        "lwin": "10127812011",
                        "lwinName": "Chateau Margaux Premier Cru Classe, Margaux",
                        "vintage": "2011",
                        "indexName": null,
                        "value": 1000.0,
                        "monthOnMonthChange": null,
                        "yearToDateChange": -2.34,
                        "oneYearChange": 4.39,
                        "twoYearChange": 2.96,
                        "fiveYearChange": 25.0
                    },
                    {
```

```
                        "lwin": "10140332009",
                        "lwinName": "Petrus, Pomerol",
                        "vintage": "2009",
                        "indexName": null,
                        "value": 1000.0,
                        "monthOnMonthChange": null,
                        "yearToDateChange": 5.97,
                        "oneYearChange": 8.56,
                        "twoYearChange": 4.68,
                        "fiveYearChange": 33.96
                    }
                ]
            },
            {
                "group": "internalIndex",
                "list": [
                    {
                        "lwin": null,
                        "lwinName": null,
                        "vintage": null,
                        "indexName": "Liv-ex Fine Wine 50",
                        "value": 378.25,
                        "monthOnMonthChange": null,
                        "yearToDateChange": 8.88,
                        "oneYearChange": 12.08,
                        "twoYearChange": 8.57,
                        "fiveYearChange": -2.53
                    },
                    {
                        "lwin": null,
                        "lwinName": null,
                        "vintage": null,
                        "indexName": "Liv-ex Fine Wine 100",
                        "value": 215.0,
                        "monthOnMonthChange": -44.75,
                        "yearToDateChange": -32.6,
                        "oneYearChange": -28.84,
                        "twoYearChange": -30.66,
                        "fiveYearChange": -20.1
                    }
                ]
            },
      ]
    },
    "errors": null
}
```


Invalid JSON response


```
{
    "status": "Unauthorized",
    "httpCode": "401",
    "message": "Request was unsuccessful",
    "internalErrorCode": "R000",
    "apiInfo": {
        "version": "1.0",
        "timestamp": 1629195812872,
        "provider": "Liv-ex"
    }
}
```


XML Response

The response is sent per request. The example below shows the response for the XML Request example.

```
<?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>2021-08-17T10:22:26.419Z</Timestamp>
        <Provider>Liv-ex</Provider>
    </ApiInfo>
    <chartDataPerformance>
        <currency>gbp</currency>
        <groups>
            <group>lwin</group>
            <list>
                <lwin>10127812010</lwin>
                <lwinName>Chateau Margaux Premier Cru Classe, Margaux</lwinName>
                <vintage>2010</vintage>
                <indexName xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:nil="true"/>
                <value>1000.0</value>
                <monthOnMonthChange xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:nil="true"/>
                <yearToDateChange>10.26</yearToDateChange>
                <oneYearChange>6.78</oneYearChange>
                <twoYearChange>0.8</twoYearChange>
                <fiveYearChange>5.74</fiveYearChange>
            </list>
            <list>
                <lwin>10127812009</lwin>
                <lwinName>Chateau Margaux Premier Cru Classe, Margaux</lwinName>
                <vintage>2009</vintage>
                <indexName xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:nil="true"/>
                <value>1000.0</value>
                <monthOnMonthChange xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:nil="true"/>
                <yearToDateChange>5.08</yearToDateChange>
                <oneYearChange>12.19</oneYearChange>
                <twoYearChange>5.08</twoYearChange>
                <fiveYearChange>10.16</fiveYearChange>
            </list>
        </groups>
        <groups>
            <group>internalIndex</group>
            <list>
                <lwin xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
                <lwinName xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:nil="true"/>
                <vintage xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
                <indexName>Liv-ex Fine Wine 50</indexName>
                <value>378.25</value>
                <monthOnMonthChange xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:nil="true"/>
                <yearToDateChange>8.88</yearToDateChange>
                <oneYearChange>12.08</oneYearChange>
                <twoYearChange>8.57</twoYearChange>
                <fiveYearChange>-2.53</fiveYearChange>
            </list>
        </groups>
    </chartDataPerformance>
    <errors xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
</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>2021-08-17T11:19:41.422+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
V006	Invalid LWIN number
V125	Requests with multiple LWINs must contain LWINs of the same length.
V169	Invalid / incorrect internalIndex name: [%s].

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.