



THE FINE WINE MARKET

Search Market API v1

Document Revision 1.5
Date of Issue: 16 January 2020
Date of revision: 15 April 2020

Nick Palmer

Product Lead

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 Search Market service (POST method).....	5
6. Response Codes.....	14
6.1 Request validation error codes	14
6.2 HTTP Status codes	15

1. Purpose

To provide the API end point information and examples of the web services available for Search Market.

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.
Wine	The word wine below is referring to a specific wine (the producer and brand, grape or vineyard), vintage and unit size combination.
Bid	A buyer places a bid on the Exchange for buying a certain amount of wine.
Offer	A seller places an offer on the Exchange for selling a certain amount of wine.
List price	A stock listing from a current price list collected by Liv-ex in the last 35 days.
Order	Order is a generic term for both bid/offer.
Market Price	The Market Price is based on the cheapest 6 and 12-pack prices advertised by leading merchants in the EU and Switzerland. (Where appropriate, alternative unit sizes are used for the calculation.) It provides a guide as to the price you are likely to pay for SIB-compliant stock in the market
SIB	Standard in Bond trade terms (link)
SEP	Standard En Primeur trade terms (link)
Special	Special contract trade terms (link)
Contract Type	Contract type is a generic term for SIB, SEP or Special (X).
LWIN18 contract market	A specific market on the exchange. A wine of a specific vintage and pack/bottle format being sold under a declared contract. e.g. Lafite 2010 12x75 SIB e.g. Sassicaia 1999 6x75 Special Now

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 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.

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",
  "statusCode": "401",
  "message": "Request was unsuccessful",
  "livexCode": "R000"
  "apiInfo": {
    "version": "4.0",
    "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>4.0</Version>
    <Timestamp>2017-11-04T11:12:30</Timestamp>
    <Provider>Liv-ex</Provider>
  </ApiInfo>
</Response>
```

5. API Listing

5.1 Search Market service (POST method)

Description

This service will be used to request live exchange information. It allows the caller to search by live bids and offers, contract type, price range, format size, origin, colour and LWIN.

Base URI

[search/v1/searchMarket](https://api.liv-ex.com/search/v1/searchMarket)

URI Pagination Parameters

Name	Mandatory	Description
?limit	N default = 25	Number of results to include per response. A "result" is counted as one LWIN18 market. Values: '25' (minimum), '50', '100', '250' (maximum) Type: integer
?offset	N default = 1	The # of the response page to return. Starting value always = 1. Type: integer

Example URI including pagination

<https://api.liv-ex.com/search/v1/searchMarket?limit=250&?offset=1>

Where a request returns more than a single page of results, consuming systems should iterate the offset value until all results have been received (a sequence of POSTs with the body remaining unchanged).

Note: During particularly busy periods of activity on the exchange results can change between submission of offset values.

Request Parameters

Name	Mandatory	Description
currency	Y default = gbp	Values: 'gbp', 'eur', 'chf', 'usd', 'hkd', 'jpy', 'sgd', 'gbpbtt', 'eurbtt', 'chfbtt', 'gbp/btt', 'eur/btt', 'chf/btt' Type: alphanumeric
lwin	N	LWIN7, LWIN11, LWIN16 or LWIN18 Type: array
vintage	N	Type: array <i>Mandatory if "lwin" length = LWIN7</i> <i>Ignored if "lwin" length is LWIN11 or greater</i>
lwinGeographic	N	Product geographic origin filter (can relate to country, region or sub-region). Any value allowed.

		Type: alphanumeric
lwinColour	N	Values: 'red', 'white', 'rose', 'na' Type: alphanumeric
priceType	N	Values: 'bid', 'offer' Type: alphanumeric array
contractType	N	Values: 'sib', 'sep', 'x' Type: alphanumeric array
dutyPaid	N (mandatory if contractType = x)	States whether the stock offered on the Special contract has a tax status of duty paid or not. If set to 'false' stock should be considered In Bond (IB). Type: Boolean 'true' or 'false'
minimumQty	N (mandatory if contractType = x)	States whether the seller has placed a minimum volume of units on the trade. E.g the seller has 50 units on offer with a minimumQty value of 10. Type: integer
deliveryPeriod	N (mandatory if contractType = x)	States whether the lead time on the offer is different to the standard Liv-ex terms of 2 weeks. If deliveryPeriod = 0, the offer is Special Now i.e. the stock is in the Liv-ex warehouse, has been checked and is ready for immediate dispatch. Type: integer
condition	N (mandatory if contractType = x)	A free text field that states any issues with the stock or its packaging. Enter a single space character (" " or <condition> </condition>) to return records with no condition issues. Type: string
packSize	N	Type: integer(2) array
bottleSize	N	Type: integer(5) array
combinedFormat	N	Can be used instead of "packSize" and "bottleSize" to specify one or more defined formats. Values must follow LWIN packSize and bottleSize syntax [2-digit]x[5-digit] e.g. "06x00750" Type: alphanumeric array
minOrderQty	N	Used to set a minimum volume of stock Type: integer
minMarketPrice	N	Type: integer
maxMarketPrice	N	Type: integer
minLastTrade	N	Type: integer
maxLastTrade	N	Type: integer
minPrice	N	Type: integer
maxPrice	N	Type: integer
isCompetitive	N	Only show bids and offers that are at a competitive

	Default = false	price point. Used by the Liv-ex exchange UI. When set to 'true' the API filters results to show - bids above Market Price - offers below or within 5% of Market Price When set to 'false' the response attribute "isCompetitive" always returns null Type: Boolean 'true' 'false'
--	-----------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Sample Request Body

JSON #1 - all live offers, currency GBP

```
{
  "currency": "gbp",
  "priceType": ["offer"]
}
```

JSON #2 - all live offers, Ch. Lafite 2008-2017, currency EUR/btt

```
{
  "currency": "eurbtt",
  "priceType": ["offer"],
  "lwin": ["1011872"],
  "vintage":
  ["2017", "2016", "2015", "2014", "2013", "2012", "2011", "2010", "2009", "2008"]
}
```

JSON #3 – all live bids on wines from Pomerol, market price < £5000 per 9l, GBP

```
{
  "currency": "GBP",
  "priceType": ["bid"],
  "lwinGeographic": "pomerol",
  "maxMarketPrice": 5000
}
```

JSON #4 – all live Special now offers from Tuscany, GBP

```
{
  "currency": "GBP",
  "priceType": ["offer"],
  "lwinGeographic": "tuscany",
  "contractType": ["x"],
  "dutyPaid": false,
  "minimumQty": "1",
  "deliveryPeriod": "0",
  "condition": " "
}
```

XML #1 – all live offers, currency GBP

```
<SearchMarket>
  <currency>GBP</currency>
  <priceType>offer</priceType>
</SearchMarket>
```

XML #2 - all live offers, Ch. Lafite 2017-2008, currency EUR/btt

```
<SearchMarket>
  <currency>gbp</currency>
  <priceType>offer</priceType>
  <lwin>1011872</lwin>
```

```
<vintage>2008</vintage>  
<vintage>2009</vintage>  
<vintage>2010</vintage>  
<vintage>2011</vintage>  
<vintage>2012</vintage>  
<vintage>2013</vintage>  
<vintage>2014</vintage>  
<vintage>2015</vintage>  
<vintage>2016</vintage>  
<vintage>2017</vintage>  
</SearchMarket>
```

XML #3 – all live bids on wines from Pomerol, market price < £5000 per 9l, GBP

```
<SearchMarket>  
  <currency>eur</currency>  
  <priceType>bid</priceType>  
  <lwinGeographic>pomerol</lwinGeographic>  
  <maxMarketPrice>5000</maxMarketPrice>  
</SearchMarket>
```

XML #4 – all live Special now offers from Tuscany

```
<SearchMarket>  
  <currency>gbp</currency>  
  <priceType>offer</priceType>  
  <lwinGeographic>Tuscany</lwinGeographic>  
  <contractType>X</contractType>  
  <dutyPaid>>false</dutyPaid>  
  <minimumQty>1</minimumQty>  
  <deliveryPeriod>0</deliveryPeriod>  
  <condition> </condition>  
</SearchMarket>
```


Sample Response Body

The Search Market service will respond with HTTP Code 200 OK in a successful response. The volume of data returned will be dependent on the request attributes supplied.

Data is sorted in descending activity order. LWIN11s that have seen new bid or offers on the exchange will appear higher up the response.

If one or more valid LWINs are submitted the API will respond in the follow way:

LWIN11

Live market data for all applicable pack/bottle combinations for the requested wine and vintage.

LWIN16

Live market data for all applicable bottle sizes for the requested wine and vintage.

LWIN18

Live market data for the specified bottle and pack size for the requested wine and vintage.

Response parameters

Name	Description
lwin	LWIN11 / LWIN18 Type: integer
lwinName	The LWIN wine name description e.g. 'Grand Puy Lacoste' Type: alphanumeric
lwinCountry	The LWIN country description e.g. 'France' Type: alphanumeric
lwinRegion	The LWIN region description e.g. 'Bordeaux' Type: alphanumeric
lwinSubRegion	The LWIN sub region description e.g. 'Pauillac' Type: alphanumeric
lwinColour	The LWIN colour description e.g. 'Red' Type: alphanumeric
vintage	Type: 4-digit integer
currency	The currency format of all pricing data contained in the response Type: alphanumeric
iwp	Link to the individual wine page (IWP) of the LWIN on the Liv-ex exchange application Type: alphanumeric
packSize	The LWIN pack size description e.g. '06' Type: 2-digit integer
bottleSize	The LWIN bottle size description e.g. '00750' Type: 5-digit integer
contractType	All, SIB, SEP, X Type: alphanumeric
dutyPaid	<i>null unless contractType = X</i> States whether stock offered on Special contract has a tax status of duty paid. If 'false' stock should be considered In Bond (IB). Type: Boolean 'true' or 'false'

minimumQty	<i>null unless contractType = X</i> States whether the seller has placed a minimum volume of units on the trade. Standard Liv-ex terms (SIB) is a minimum quantity of 1. Type: integer
deliveryPeriod	<i>null unless contractType = X</i> States the lead time on the offer in weeks. Standard Liv-ex terms (SIB) are 2 weeks. If deliveryPeriod = 0, the offer is Special Now Type: integer
condition	<i>null unless contractType = X</i> Text field that states any issues with the stock or its packaging. Type: string
photold	The photo resource ID Type: integer
lowResUrl	HTTP link to a low resolution version of the photo resource Type: alphanumeric
highResUrl	HTTP link to a high resolution version of the photo resource Type: alphanumeric
parentOrderGUID	<i>null unless contractType = x and priceType = bid</i> The orderGUID reference of the offer that the bid position is linked to Type: 128-bit hexadecimal
marketPrice	Type: double
lastTradePrice	Type: double
lastTradeDate	Type: date (epoch time if JSON)
priceType	Values: 'bid', 'offer', 'list' Type: string
price	Type: double
quantity	Type: integer
isBest	The position is the leading live bid/offer for the specific product and contract type. For priceType = 'list' this attribute will show as null. Type: Boolean true/false
myPosition	The bid/offer position is owned by the merchant calling the service. Type: Boolean true/false
orderGUID	The GUID of the bid/offer position. For priceType = 'list' this attribute will show as null. Type: 128-bit hexadecimal
priceDate	Type: date (epoch time if JSON)
isCompetitive	Declares if the position is above Market Price (priceType = bid), or within or below 5% of Market price (priceType = offer) Type: Boolean 'true' 'false'

JSON Response

The response is sent per request. The example below shows one live SIB market for Lafite 2012 12x75 and features one live bid position, one live offer position, and one list price.

```
{
  "pageInfo": {
    "totalResults": 1,
    "limit": 25,
    "offset": 1
  },
  "status": "OK",
  "statusCode": "200",
  "message": "Request completed successfully",
  "internalErrorCode": "R001",
  "apiInfo": {
    "version": "1.0",
    "timestamp": 1577181294193,
    "provider": "Liv-ex"
  },
  "searchResponse": [
    {
      "lwin": "10118722012",
      "lwinName": "Lafite Rothschild",
      "lwinCountry": "France",
      "lwinRegion": "Bordeaux",
      "lwinSubRegion": "Pauillac",
      "lwinColour": "Red",
      "vintage": "2012",
      "currency": "GBP",
      "iwp": "https://tpg-qa-app.liv-ex.com/#/wine-page?lwin11=10118722012",
      "market": [
        {
          "lwin": "101187220121200750",
          "packSize": "12",
          "bottleSize": "00750",
          "contractType": "SIB",
          "special": {
            "dutyPaid": false,
            "minimumQty": 1,
            "deliveryPeriod": 2,
            "condition": null,
            "photos": [],
            "parentOrderGUID": null
          },
          "marketPrice": 5100.0,
          "lastTradePrice": 4780.0,
          "lastTradeDate": "2019-05-08T15:39:49.000+0000",
          "depth": {
            "bids": {
              "bid": [
                {
                  "priceType": "Bid",
                  "price": 300.0,
                  "quantity": 1,
                  "isBest": true,
                  "myPosition": true,
                  "orderGUID": "cf45a9ed-1e35-40c9-a6ef-8de1ddb8a9e3",
                  "priceDate": "2019-12-02T10:30:19.000+0000",
                  "isCompetitive": null
                }
              ]
            }
          },
          "offers": {
            "offer": [
              {
                "priceType": "Offer",
                "price": 4700.0,
                "quantity": 1,
                "isBest": true,
                "myPosition": true,
            }
          ]
        }
      ]
    }
  ]
}
```

```

        "orderGUID": "3a690b6d-2813-4564-95cf-5277ea307df4",
        "priceDate": "2019-11-11T09:09:27.000+0000",
        "isCompetitive": null
      }
    ]
  },
  "lists": {
    "list": [
      {
        "priceType": "List",
        "price": 2220.0,
        "quantity": 2,
        "isBest": true,
        "myPosition": false,
        "orderGUID": null,
        "priceDate": "2019-11-26",
        "iscompetitive": null
      }
    ]
  }
}
  ],
  "errors": null
},
"errors": null
}

```

Invalid JSON response

```

{
  "pageInfo": null,
  "status": "Bad Request",
  "statusCode": "400",
  "message": "Request was unsuccessful",
  "internalErrorCode": "R000",
  "apiInfo": {
    "version": "1.0",
    "timestamp": 1577182193363,
    "provider": "Liv-ex"
  },
  "searchResponse": null,
  "errors": {
    "error": [
      {
        "code": "V002",
        "message": "Invalid parameter(lwin).\"
      }
    ]
  }
}

```

XML Response

The response is sent per request. The example below shows one live SIB market for Lafite 2012 12x75 and features one live bid position, one live offer position, and one list price.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<searchMarketResponses>
  <Status>OK</Status>
  <HttpCode>200</HttpCode>
  <Message>Request completed successfully</Message>
  <InternalErrorCode>R001</InternalErrorCode>
  <ApiInfo>

```

```

<Version>1.0</Version>
<Timestamp>2019-12-24T10:11:13.023Z</Timestamp>
<Provider>Liv-ex</Provider>
</ApiInfo>
<pageInfo>
  <totalResults>1</totalResults>
  <limit>25</limit>
  <offset>1</offset>
</pageInfo>
<searchResponse>
  <lwin>10118722012</lwin>
  <lwinName>Lafite Rothschild</lwinName>
  <lwinCountry>France</lwinCountry>
  <lwinRegion>Bordeaux</lwinRegion>
  <lwinSubRegion>Pauillac</lwinSubRegion>
  <lwinColour>Red</lwinColour>
  <vintage>2012</vintage>
  <currency>GBP</currency>
  <iwp>https://tpg-qa-app.liv-ex.com/#/wine-page?lwin11=10118722012</iwp>
  <market>
    <lwin>101187220121200750</lwin>
    <packSize>12</packSize>
    <bottleSize>00750</bottleSize>
    <contractType>SIB</contractType>
    <special>
      <dutyPaid>>false</dutyPaid>
      <minimumQty>1</minimumQty>
      <deliveryPeriod>2</deliveryPeriod>
    </special>
    <marketPrice>5100.0</marketPrice>
    <lastTradePrice>4780.0</lastTradePrice>
    <lastTradeDate>2019-05-08T15:39:49Z</lastTradeDate>
    <depth>
      <bids>
        <bid>
          <priceType>Bid</priceType>
          <price>300.0</price>
          <quantity>1</quantity>
          <isBest>true</isBest>
          <myPosition>true</myPosition>
          <orderGUID>cf45a9ed-1e35-40c9-a6ef-8de1ddb8a9e3</orderGUID>
          <priceDate>2019-12-02T10:30:19Z</priceDate>
          <isCompetitive>false</iscompetitive>
        </bid>
      </bids>
      <offers>
        <offer>
          <priceType>Offer</priceType>
          <price>4700.0</price>
          <quantity>1</quantity>
          <isBest>true</isBest>
          <myPosition>true</myPosition>
          <orderGUID>3a690b6d-2813-4564-95cf-5277ea307df4</orderGUID>
          <priceDate>2019-11-11T09:09:27Z</priceDate>
          <isCompetitive>false</iscompetitive>
        </offer>
      </offers>
      <lists>
        <list>
          <priceType>List</priceType>
          <price>2220.0</price>
          <quantity>2</quantity>
          <isBest>true</isBest>
          <myPosition>>false</myPosition>
          <priceDate>2019-11-26T00:00:00Z</priceDate>
        </list>
      </lists>
    </depth>
  </market>
</searchResponse>

```

```

        </lists>
    </depth>
</market>
</searchResponse>
</searchMarketResponses>

```

Invalid XML Response

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<searchMarketResponses>
  <Status>Bad Request</Status>
  <HttpCode>400</HttpCode>
  <Message>Request was unsuccessful</Message>
  <InternalErrorCode>R000</InternalErrorCode>
  <ApiInfo>
    <Version>1.0</Version>
    <Timestamp>2019-12-24T10:12:55.769Z</Timestamp>
    <Provider>Liv-ex</Provider>
  </ApiInfo>
  <errors>
    <error>
      <code>V002</code>
      <message>Invalid parameter(lwin).</message>
    </error>
  </errors>
</searchMarketResponses>

```

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
V001	Mandatory field missing
V002	Merchant is not allowed to access the requested feed
V003	Invalid parameter(s)
V005	Invalid number parameter: positive number expected for <paramName>
V006	Merchant is not active
V013	Invalid request headers. Please provide value for header <header name>
V058	Invalid / incorrect LWIN: [{v}]. Please provide a valid LWIN11, LWIN16 or LWIN18 code.

V061	Invalid / incorrect currency: [<value>]. Possible values are 'gbp', 'eur', 'chf', 'usd', 'hkd', 'jpy', 'sgd', 'gbp/btt', 'eur/btt', 'chf/btt', 'gbpbtt', 'eurbtt', 'chfbtt'."
V073	Invalid / incorrect dutyPaid: [%s]. Possible values are 'true', 'false'
V074	Invalid / incorrect deliveryPeriod: [%1\$s]. Must be positive integer value. Max value = %2\$s weeks
V075	Invalid / incorrect minimumQty: [%s]. Must be a positive integer value
V077	Invalid / incorrect contractType: [%s]. Possible values can be 'sib' (Standard In Bond), 'sep' (Standard En Primeur) and 'x' (Special).
V125	Requests with multiple LWINs must contain LWINs of the same length.
V126	For LWIN7 requests one or more vintage values must be provided.
V127	Invalid / incorrect combinedFormat value. Expected format is LWIN packSize x LWIN bottleSize e.g "06x00750".
V128	Invalid / incorrect isCompetitive: [%s]. Possible values are 'true' or 'false'.
V129	Invalid / incorrect priceType: [%s]. Possible values are 'bid', 'offer', 'list', 'marketPrice' and 'lastTrade'.

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.