LIV|EX
THE FINE WINE MARKET

Order by UID API v1

Document revision 1.2
Date of Issue: 04 October 2018
Date of revision: 10 May 2021

Nick Palmer

Product Manager

Table of Contents

1. Purpose

To provide the API endpoint information and examples of the web services available for the Order by UID service.

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.
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
Contract Type	Contract type is a generic term for SIB, SEP or Special (X).
SIB	Standard in Bond trade terms (link)
SEP	Standard En Primeur trade terms (link)
Special (X)	Special contract trade terms (link)
Special Now	An offer of stock that is ready for immediate dispatch from Liv-ex warehouses
UID	A unique identifier assigned to each and every case of wine stored in Liv-ex warehouses

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's will support the following method(s):
    1. POST (request parameter(s) are contained in message body)
- 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 Order by UID API will be accessible at https://api.liv-ex.com/

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.

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": "Request was unsuccessful",
    "livexCode": "R000"
    "apiInfo": {
        "version": "1.0",
        "timestamp": 1555595165883,
        "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>2019-04-18T11:12:30</Timestamp>
        <Provider>Liv-ex</Provider>
    </ApiInfo>
<Response>
```

5. API Listing

5.1 Order by UID Service (POST method)

Description

This service may be used by merchant systems to place stock held in Liv-ex warehouses onto the exchange. It facilitates Special Now offers to be created via API.

In contrast to the Orders API where LWIN and contract details must be specified, the Order By UID API uses Vine UID metadata to generate the product (LWIN) and contract terms for each order. If a UID is Special Now compliant, that order will benefit from an accelerated settlement process once traded.

The service accepts one or more order UIDs per request (up to a maximum limit of 50 per request). To successfully process multiple UIDs they must have the same:

1. LWIN18
2. Warehouse location (TIL)
3. Warehouse status (A)
4. Paid status (paid)

The API permits multiple UIDs with differing SIB passport statuses, duty statuses, condition issues and photos to be submitted. So long as the 4 points above are consistent, Liv-ex systems will split the UIDs into appropriate contract groups based on the UID metadata.

Note that UIDs without an approved SIB Passport must have a visible and valid (no more than 3 years old) photo in order to be sold as Special Now. A sister API service, **availableNow**, can be used to query for these UIDs.

To update or delete an order placed by the Order By UID service, PATCH and DELETE commands should be used via the Orders API.

Base URI

/exchange/v1/orderByUID

Param

Name	Mandatory	Description
UID	Y	A Liv-ex warehouse unique identification code (UID). Multiple values are permitted if placing offers for multiple cases of a given LWIN18

		Type: a 3-digit to 7-digit integer value
orderStatus	Y	A valid order status. The possible values can be L (live) and S (suspended). See recommendation note. **Type:** alphanumeric
expiryDate	N	A valid future date in yyyy-mm-dd format e.g. 2015-07-31. **Type:** alphanumeric, ISO8601 format
currency	Y	Either EUR or GBP. The currency used must match the trading currency pre-agreed with Liv-ex. **Type:** 3-character alphanumeric
price	Y	A valid positive value. GBP prices will be rounded to the nearest whole integer. EUR prices will be round to 1 decimal place. **Type:** integer or double
merchantRef	N	An optional text field used to attach a reference to the order. Limited to 30 characters. Strings exceeding this limit will be truncated. **Type:** alphanumeric (30-character limit)
enforcePhoto	N	An optional Boolean field that, when set to true, will force any optional valid photos to be linked to the offer on Liv-ex. **Note:** offers with photos enforced will display as "Special – Condition". For stock where photos are mandatory, images will be linked irrespective of the value of this field. **Type:** Boolean true/false (default false)

Sample request body

JSON Request

Single UID

```
{
  "UID": ["1386411"],
  "orderStatus": "L",
  "expiryDate": 1549537950898,
  "currency": "GBP",
  "price": "980",
  "merchantRef": "broking_case_Mr_Smith",
  "enforcePhoto": true
}
```

Multiple UIDs

```
{
  "UID": ["1386411", "1386412","1386413"],
  "orderStatus": "L",
  "expiryDate": 1549537950898,
```

```
  "currency": "GBP",
  "price": "980",
  "merchantRef": "order_by_uid_test"
}
```

XML Request

Single UID

```
<OrderByUID>
  <UID>1386411</UID>
  <orderStatus>L</orderStatus>
  <expiryDate>2018-11-05</expiryDate>
  <currency>GBP</currency>
  <price>980</price>
  <merchantRef> broking_case_Mr_Smith </merchantRef>
  <enforcePhoto>true</enforcePhoto>
</OrderByUID>
```

Multiple UIDs

```
<OrderByUID>
  <UID>1386411</UID>
  <UID>1386412</UID>
  <UID>1386413</UID>
  <orderStatus>L</orderStatus>
  <expiryDate>2018-11-05</expiryDate>
  <currency>GBP</currency>
  <price>980</price>
  <merchantRef>order_by_uid_test</merchantRef>
</OrderByUID>
```

JSON response

The response is sent per request

```
{
  "status": "OK",
  "httpCode": "200",
  "message": "Request completed successfully.",
  "internalErrorCode": "R001",
  "apiInfo": {
    "version": "1.0",
    "timestamp": 1537451805694,
    "provider": "Liv-ex"
  },
  "orders": {
    "order": [
      {
        "merchantRef": "order_by_uid_test",
        "orderGUID": "c2820e23-f3b4-4974-bfca-a07f074ca879",
        "orderPlaceDate": 1537451805607,
        "errors": null
      }
    ]
  }
}
```

Invalid JSON response

```
{
  "status": "Multiple statuses",
  "httpCode": "207",
  "message": "Request partially completed",
  "internalErrorCode": "R002",
```

```
  "apiInfo": {
    "version": "1.0",
    "timestamp": 1537453142302,
    "provider": "Liv-ex"
  },
  "orders": {
    "order": [
      {
        "merchantRef": null,
        "orderGUID": null,
        "orderPlaceDate": null,
        "errors": {
          "error": [
            {
              "code": "V083",
              "message": "UID is already being offered on Liv-ex"
            }
          ]
        }
      }
    ]
  }
}
```


XML response

The response is sent per request


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<exchangeResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="https://qa-api.liv-ex.com/v1 https://qa-api.liv-ex.com/schema/v1/services.xsd">
  <Status>OK</Status>
  <HttpCode>200</HttpCode>
  <Message>Request completed successfully.</Message>
  <InternalErrorCode>R001</InternalErrorCode>
  <ApiInfo>
    <Version>1.0</Version>
    <Timestamp>2018-09-20T15:19:53.627+01:00</Timestamp>
    <Provider>Liv-ex</Provider>
  </ApiInfo>
  <Orders>
    <order>
      <MerchantRef>order_by_uid_test</MerchantRef>
      <OrderGUID>96816d22-e94a-4256-b097-d82ae10e9b4c</OrderGUID>
      <OrderPlaceDate>2018-09-20T15:19:53.527+01:00</OrderPlaceDate>
      <Errors xsi:nil="true"/>
    </order>
  </Orders>
</exchangeResponse>
```


Invalid XML response


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<exchangeResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="https://qa-api.liv-ex.com/v1 https://qa-api.liv-ex.com/schema/v1/services.xsd">
  <Status>Multiple statuses</Status>
  <HttpCode>207</HttpCode>
  <Message>Request partially completed</Message>
  <InternalErrorCode>R002</InternalErrorCode>
  <ApiInfo>
    <Version>1.0</Version>
```

```
    <Timestamp>2018-09-20T15:15:55.162+01:00</Timestamp>
    <Provider>Liv-ex</Provider>
  </ApiInfo>
  <Orders>
    <order>
      <MerchantRef xsi:nil="true"/>
      <OrderGUID xsi:nil="true"/>
      <OrderPlaceDate xsi:nil="true"/>
      <Errors>
        <error>
          <code>V083</code>
          <message>UID is already being offered on Liv-ex</message>
        </error>
      </Errors>
    </order>
  </Orders>
</exchangeResponse>
```

Response Parameters

Name	Description
merchantRef	The optional text field that was included as part of the initial request.
orderGUID	The GUID of the new order that has been created. The GUID is required when sending a DELETE request.
orderPlaceDate	The timestamp of when the new order was placed.

6. Exchange validations override

Description

The Liv-ex exchange platform validates every order it receives to ensure the correct products are traded. A subset of validations (known internally as 'fat finger') look for price and quantity values differ significantly from normal and block these offers.

- **Fat finger weak / strong**
  Bids that are significantly higher than the market price or offers that are significantly lower than the market price are rejected.
- **Quantity higher than price**
  Orders where the quantity value is higher than the sale price per unit are rejected.

It is possible to opt out of these validations. Please contact your exchange manger if you would like to know more.

7. Supplementary API services

orders API – place bids and offers on the exchange

heartbeat API – check the exchange is available

orderStatus API – check the status of specific bid/offer positions on the exchange

myPositions API – view and reconcile all your positions with Liv-ex (live or suspended)

bulkOrderAction API – suspend, reactivate and renew positions on the exchange in bulk

8. 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

8.1 Request validation error codes

Code	Message
V000	Mandatory field missing.
V001	Merchant is not allowed to access the requested feed.
V002	Invalid parameter(s).
V003	Wrong date format. Date should be 'yyyy-MM-dd'.
V004	Invalid number parameter: positive number expected for {paramName}.
V005	Merchant is not active.
V006	Invalid LWIN number.
V007	Invalid LWIN 7.
V008	Invalid LWIN 18.
V009	Web service only supports B (Bid) and O (Offer) as order type parameter.
V010	Web service only supports SIB and SEP as contract type parameter.
V011	Web service only supports L (Live) and S (Suspend) as order state parameter.
V012	Invalid request headers. Please provide value for header {header name}.
V013	Please provide valid vintage.
V015	Invalid currency.
V017	Merchant does not have the EP limit assigned for vintage <value>
V018	Mandatory field missing (<value>)
V053	GUID is mandatory for contract type X.
V054	Parent order is not live

V055	Order details do not match order GUID
V056	GUID is not available or does not exist
V072	Quantity change is not allowed in this order.
V073	Invalid / incorrect dutyPaid: [<value>]. Possible values are 'true', 'false'
V074	Invalid / incorrect deliveryPeriod: [<value>]. Must be positive integer value. Max value = 16 weeks
V075	Invalid / incorrect minimumQty: [<value>]. Must be a positive integer value
V077	Invalid / incorrect contractType: [<value>]. Possible values can be 'sib' (Standard In Bond), 'sep' (Standard En Primeur) and 'x' (Special).
V078	Missing attribute: [<value>]. When orderType = 'o' and contractType = 'x', attributes dutyPaid, minimumQty and deliveryPeriod must be included.
V080	Invalid / incorrect uids: [<value>]. Must be a positive integer value
V081	Invalid / incorrect enforcePhoto: [<value>]. Possible values are 'true', 'false'
V082	UID properties are not the same - the order has not been placed. [<value>] must be the same across all UIDs. Please check the UIDs submitted or send as individual orders.
V083	UID is already being offered on Liv-ex
V084	If specialOrderGUID is supplied contractType must be 'X'.
V085	Parent and child orders can't have same order type.
V086	Please provide valid special terms of contract to create a special order
V087	Contract type change is not allowed in this order.

8.2 Trade validation error codes

Code	Error Key	Meaning
TR001	invalid.made.avail	The delivery period supplied is not valid. Must be a positive integer value (max value = 16)
TR002	invalid.min.unit.and.qty	Your bid does not meet the minimum quantity terms of the contract
TR003	bid.fat.finger.weak.warn	A bid appears to be above Market Price. Please check carefully before proceeding.
TR004	bid.fat.finger.strong.warn	A bid appears to be above Market Price. Please check carefully before proceeding.
TR005	offer.fat.finger.weak.warn	An offer appears to be below Market Price. Please check carefully before proceeding.
TR006	offer.fat.finger.strong.warn	An offer appears to be below Market Price. Please check carefully before proceeding.

TR007	order.did.not.confirm. successfully.try.again	The order did not confirm successfully. Please check request syntax and try again.
TR010	phy.offer.restrict.due.to. exceeding.qty.than.stored	Merchant trading status is set to 'buy and resell only'.
TR011	merchant.about.to.match.his. offer	Merchant is about to match their own offer
TR012	merchant.about.to.match.his. bid	Merchant is about to match their own bid
TR014	merchant.status.no.trading	Merchant does not have trading privileges.
TR015	merchant.status.sell.only.no. phy.bid	Merchant account is ‘sell only'.
TR016	merchant.status.sell.only.ep. resell.for.a.vintage	Merchant is not permitted to buy EP stock. EP sell status is ‘resell only’.
TR017	merchant.status.sell.only.no. ep.for.a.vintage	Merchant is not permitted to sell EP stock.
TR018	merchant.status.sell.only.ep. allowed	Merchant is not permitted to sell EP stock.
TR019	merchant.status.sso.no.ep.for .a.vintage	Merchant is not allowed to buy and sell EP stock as the trading status is ‘sell special only.
TR020	sso.offer.restrict.due.to.exceed ing.qty.than.stored	You are currently only allowed to create special offers. Please change your offer

8.3 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.

9. Appendix – Special contracts types

Special contracts (contractType = 'X') carry four attributes that define the tax status, minimum volume, lead time and condition of a specific offer. Attributes can be combined in various ways depending on the status of the stock.

Special attribute	Meaning
dutyPaid	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	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	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 n the Liv-ex warehouse, has been checked and is ready for immediate dispatch. **Type:** integer
condition	A free text field that states any issues with the stock or its packaging. **Type:** string

Some wine offered under a Special contract can match or exceed the Liv-ex SIB terms. Offers listed as 'Special – Now' on the exchange are the equivalent of Standard In Bond (SIB) but have the added benefit of being ready for immediate dispatch from Liv-ex warehouses.

The following combination of attributes would filter to these specific type of Special offers:

- dutyPaid = false
- minimumQty = null
- deliveryPeriod = 0
- condition = null

Offers with these flags are In Bond (IB), have no minimum quantity terms or condition issues and have already been landed and checked in the Liv-ex warehouses.