



Bid Offer Change Since API v1

Document Revision 1.1
Date of Issue: 22 March 2021
Date of revision: 21 May 2021

Table of Contents

1. Purpose	3
2. Glossary of Terms	3
3. Technical Standards	3
4. Request Header	3
5. API Listing	4
5.1 Bid Offer Change Since service (POST method)	4
6. Response Codes	10
6.1 Request validation error codes	11
6.2 HTTP Status codes	11

1. Purpose

To provide the API end point information and examples of the web services available for Bid Offer Change Since.

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.

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 service supports ISO 8601.
- The service only support HTTPS protocol for client and server communications.
- The API will support the following methods:
 - POST for read 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.

Parameter

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

		<p>The values for the content type will be application/json or application/xml.</p> <p>If no/invalid content type is found in the request, then JSON format will be used by default.</p>
CONTENT-TYPE	Y for POST requests	<p>Content-type is a way to specify the media type of request being sent from the client to the server.</p> <p>The values for the content type will be application/json or application/xml.</p> <p>If no/invalid content type is found in the request, then JSON format will be used by default.</p>

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",
  "statusCode": "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 Bid Offer Change Since service (POST method)

Description

This service allows users to request information about live bids and offers changes for a specified timeframe. The API service returns information about the event, timestamp of the event and information about the position.

Base URI

</exchange/v1/bidOfferChangeSince>

URI Pagination Parameters

Name	Mandatory	Description
?limit	N default = 50	Values: Any value between '1' and '250' Type: integer
?offset	N default = 1	Type: alphanumeric

Example base URI including pagination: </exchange/v1/bidOfferChangeSince?offset=1&limit=25>

Request Parameters

Name	Mandatory	Description
timeframe	N Default = 5 minute	Values: '1minute', '5minute', '30minute', '1hour', '12hour', '24hour', '48hour' Type: alphanumeric <i>If both timeframe and changeSince values are provided, changeSince will be ignored.</i>
changeSince	N	Type: time YYYY-MM-DD HH:MI Maximum 48 hours <i>If the user requests changes for longer than 48h period, the API will return updates for the last 48h only</i>
priceType	N	Values: 'bid', 'offer' Type: array
contractType	N	Values: 'SIB', 'SEP', 'X' Type: array
currency	N Default = gbp	Values: 'gbp', 'eur', 'chf', 'usd', 'hkd', 'jpy', 'sgd', 'gbp/btt', 'eur/btt', 'chf/btt' Type: alphanumeric

Sample Request Body

JSON Request

```
{
  "bidOfferChangeSince": {
    "changeSince": "2021-02-22 15:30",
    "priceType": ["offer", "bid"],
    "contractType": ["SIB"],
    "currency": "EUR"
  }
}
```

XML Request

```
<bidOfferChangeSinceRequest>
  <bidOfferChangeSince>
    <currency>GBP</currency>
    <priceType>offer</priceType>
    <contractType>SIB</contractType>
    <timeframe>12hour</timeframe>
  </bidOfferChangeSince>
</bidOfferChangeSinceRequest>
```

Sample Response Body

The Bid Offer Change Since service will respond with HTTP Code 200 OK in a successful response.

Response parameters

Name	Description
changeType	Values: offerNew, bidNew, offerUpdate, bidUpdate, offerDeletion, bidDeletion, offerBecameBest, bidBecameBest Type: alphanumeric
orderGUID	The GUID of the bid/offer position. Type: 128-bit hexadecimal
priceType	Values: 'bid', 'offer' Type: alphanumeric
changeDate	Type: date (epoch time if JSON)
lwin	LWIN18 Type: alphanumeric
lwinName	The LWIN vintage specific display name description e.g. 'Chateau Grand-Puy-Lacoste 5eme Cru Classe, Pauillac' 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: alphanumeric
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: alphanumeric
bottleSize	The LWIN bottle size description e.g. '00750' Type: alphanumeric
contractType	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. Type: integer
condition	<i>null unless contractType = X</i> Text field that states any issues with the stock or its packaging. Type: alphanumeric
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
price	Type: double
quantity	Type: integer
isBest	The position is the leading live bid/offer for the specific product and contract type. Type: Boolean true/false
myPosition	The bid/offer position is owned by the merchant calling the service. Type: Boolean true/false
priceDate	Type: date (epoch time if JSON)

JSON Response

The response is sent per request.

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

```

"apiInfo": {
  "version": "1.0",
  "timestamp": 1616514619740,
  "provider": "Liv-ex"
},
"pageInfo": {
  "totalResults": 2,
  "limit": 50,
  "offset": 1
},
"bidOfferChangeSince": [
  {
    "changeType": "bidBecameBest",
    "orderGUID": "8be8c3f5-bf66-16aefef89ec8",
    "priceType": "Bid",
    "changeDate": 1616432406000,
    "orderDetails": {
      "lwin": "115731420150600750",
      "lwinName": "Giacomo Fenocchio, Barolo, Villero",
      "lwinCountry": "Italy",
      "lwinRegion": "Piedmont",
      "lwinSubRegion": "Barolo",
      "lwinColour": "Red",
      "vintage": "2015",
      "currency": "EUR",
      "iwp": "https://app.liv-ex.com/#/wine-page?lwin11=11573142015",
      "packSize": "06",
      "bottleSize": "00750",
      "contractType": "SIB",
      "special": {
        "dutyPaid": null,
        "minimumQty": null,
        "deliveryPeriod": null,
        "condition": null,
        "photos": null,
        "parentOrderGUID": null
      },
      "price": 188.76,
      "quantity": 1,
      "isBest": true,
      "myPosition": false,
      "priceDate": 1614657074000
    }
  },
  {
    "changeType": "bidDeletion",
    "orderGUID": "c25c3547-d414-4d0a-ae8e-c449a74efc5c",
    "priceType": "Bid",
    "changeDate": 1616432406000,
    "orderDetails": null
  }
],
"errors": null
}

```

Invalid JSON response

```

{
  "status": "Bad Request",
  "statusCode": "400",
  "message": "Request was unsuccessful",
  "internalErrorCode": "R000",
  "apiInfo": {
    "version": "1.0",
    "timestamp": 1613991919637,

```



```

    "provider": "Liv-ex"
  },
  "pageInfo": null,
  "bidOfferChangeSince": null,
  "errors": {
    "error": [
      {
        "code": "V164",
        "message": "Wrong changeSince format. Requested date should be a valid date in 'YYYY-MM-DD HH:mm' format."
      }
    ]
  }
}

```

XML Response

The response is sent per request.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<bidOfferChangeSinceResponse>
  <Status>OK</Status>
  <HttpCode>200</HttpCode>
  <Message>Request completed successfully</Message>
  <InternalErrorCode>R001</InternalErrorCode>
  <ApiInfo>
    <Version>1.0</Version>
    <Timestamp>2021-03-23T15:46:49.066Z</Timestamp>
    <Provider>Liv-ex</Provider>
  </ApiInfo>
  <pageInfo>
    <totalResults>2</totalResults>
    <limit>50</limit>
    <offset>1</offset>
  </pageInfo>
  <bidOfferChangeSince>
    <changeSince>
      <changeType>offerBecameBest</changeType>
      <orderGUID>b3188b5-557e65ab4389</orderGUID>
      <priceType>Offer</priceType>
      <changeDate>2021-03-22T04:02:48Z</changeDate>
      <orderDetails>
        <lwin>110768320160600750</lwin>
        <lwinName>Fonseca, Vintage Port</lwinName>
        <lwinCountry>Portugal</lwinCountry>
        <lwinRegion>Porto</lwinRegion>
        <lwinSubRegion xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
        <lwinColour xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
        <vintage>2016</vintage>
        <currency>GBP</currency>
        <iwp>https://app.liv-ex.com/#/wine-page?lwin11=11076832016</iw>
        <packSize>06</packSize>
        <bottleSize>00750</bottleSize>
        <contractType>SIB</contractType>
        <special>
          <dutyPaid xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
          <minimumQty xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>

```

```

        <deliveryPeriod xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:nil="true"/>
        <condition xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
        <parentOrderGUID xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:nil="true"/>
        </special>
        <price>275.0</price>
        <quantity>1</quantity>
        <isBest>true</isBest>
        <myPosition>false</myPosition>
        <priceDate>2021-01-11T09:05:11Z</priceDate>
    </orderDetails>
</changeSince>
<changeSince>
    <changeType>offerDeletion</changeType>
    <orderGUID>4a47e46f-9ee5-c2130cc247c4</orderGUID>
    <priceType>Offer</priceType>
    <changeDate>2021-03-22T04:02:45Z</changeDate>
    <orderDetails xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
    </changeSince>
</bidOfferChangeSince>
<errors xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
</bidOfferChangeSinceResponse>

```

Invalid XML Response

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<bidOfferChangeSinceResponse>
    <Status>Bad Request</Status>
    <HttpCode>400</HttpCode>
    <Message>Request was unsuccessful</Message>
    <InternalErrorCode>R000</InternalErrorCode>
    <ApiInfo>
        <Version>1.0</Version>
        <Timestamp>2021-02-22T11:06:05.188Z</Timestamp>
        <Provider>Liv-ex</Provider>
    </ApiInfo>
    <bidOfferChangeSince xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
    <errors>
        <error>
            <code>V077</code>
            <message>Invalid / incorrect contractType: [special]. Possible values can be
'sib' (Standard In Bond), 'sep' (Standard En Primeur) and 'x' (Special).</message>
        </error>
    </errors>
</bidOfferChangeSinceResponse>

```

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
V061	("Invalid / incorrect currency: [UR]. Possible values are 'gbp', 'eur', 'chf', 'usd', 'hkd', 'jpy', 'sgd', 'gbp/btt', 'eur/btt', 'chf/btt'.")
V077	("Invalid / incorrect contractType: [%s]. Possible values can be 'sib' (Standard In Bond), 'sep' (Standard En Primeur) and 'x' (Special).")
V127	("Invalid / incorrect priceType: [%s]. Possible values are 'bid' and 'offer'.")
V163	("Invalid / incorrect timeframe: [12hou]. Possible values are '1minute', '5minute', '30minute', '1hour', '12hour', '24hour' and '48hour'.")
V164	("Wrong changeSince format. Requested date should be a valid date in 'YYYY-MM-DD HH:mm' format.")

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.