



## Orders API v4

Document revision 4.0.1  
Date of Issue: 04 October 2018  
Date of revision: 14 November 2018

Nick Palmer  
Product Manager

## Table of Contents

|  |           |
|--|-----------|
| <b>1. Purpose</b>                            | <b>3</b>  |
| <b>2. Glossary of Terms</b>                  | <b>3</b>  |
| <b>3. Technical Standards</b>                | <b>3</b>  |
| <b>4. Request Header</b>                     | <b>4</b>  |
| <b>5. API Listing</b>                        | <b>5</b>  |
| 5.1 Add Order Service (POST method)          | 5         |
| 5.2 Edit Order Service (PATCH method)        | 13        |
| 5.3 Delete Order Service (DELETE method)     | 17        |
| <b>6. Exchange validations override</b>      | <b>19</b> |
| <b>7. Supplementary services</b>             | <b>19</b> |
| <b>8. Response Codes</b>                     | <b>20</b> |
| 8.1 Request validation error codes           | 20        |
| 8.2 Trade validation error codes             | 21        |
| 8.3 HTTP Status codes                        | 22        |
| <b>9. Appendix – Special contracts types</b> | <b>23</b> |

## 1. Purpose

To provide the API endpoint information and examples of the web services available for Exchange Integration.

## 2. Glossary of Terms

| Term                 | Meaning   |
|----------------------|---|
| <b>LWIN</b>          | 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. |
| <b>Wine</b>          | The word wine below is referring to a specific wine (the producer and brand, grape or vineyard), vintage and unit size combination.   |
| <b>Bid</b>           | A buyer places a bid on the Exchange for buying a certain amount of wine.   |
| <b>Offer</b>         | A seller places an offer on the Exchange for selling a certain amount of wine.  |
| <b>Order</b>         | Order is a generic term for both bid/offer.   |
| <b>Market Price</b>  | 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                                       |
| <b>Contract Type</b> | Contract type is a generic term for SIB, SEP or Special (X).  |
| <b>SIB</b>           | Standard in Bond trade terms ( <a href="#">link</a> )   |
| <b>SEP</b>           | Standard En Primeur trade terms ( <a href="#">link</a> )  |
| <b>Special</b>       | Special contract trade terms ( <a href="#">link</a> )   |
| <b>Special Now</b>   | An offer of stock that is ready for immediate dispatch from 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 methods:
  1. POST for create operation
  2. PATCH for making partial change to existing resource
  3. DELETE for delete operation
- POST, PATCH and DELETE services are one order at a time by default, but multiple orders and

- deletions are possible.
- Pretty printing for output readability only is supported if required
- Compression for bandwidth savings are used
- For HTTP users who can only work on GET & POST methods, we provide a Header 'X-HTTP-Method-Override' for PATCH & DELETE
- Authentication mechanism will be custom based on CLIENT\_KEY and CLIENT\_SECRET
- The Orders API will be accessible at <https://api.liv-ex.com/exchange>

## 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.<br><br>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.<br><br>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",
  "statusCode": "401",
  "message": "Request was unsuccessful",
  "livexCode": "R000"
  "apiInfo": {
    "version": "2.0",
    "timestamp": "2017-11-04T11:12:30",
    "provider": "Liv-ex"
  }
}
```

**Invalid header XML response**

```
<Response>
  <Status>Unauthorized</Status>
  <HttpCode>401</Code>
  <Message>Request was unsuccessful.</Message>
  <LivexCode>R001</LivexCode>
  <ApiInfo>
    <Version>2.0</Version>
    <Timestamp>2017-11-04T11:12:30</Timestamp>
    <Provider>Liv-ex</Provider>
  </ApiInfo>
</Response>
```

**5. API Listing**

**5.1 Add Order Service (POST method)**

**Description**

This service will be used to add bid or offer positions to the Liv-ex exchange. The Orders API enables users to create, modify and delete orders. Version 4 of the service supports all Liv-ex contract types (SIB, SEP and Specials).

Note that a successful POST request will be responded with an orderGUID value that should be recorded. This orderGUID reference can then be used in edit (PATCH) and delete (DELETE) requests to manipulate the bid/offer accordingly (an example of the response is included below).

Users integrating with the Orders API may choose to receive PUSH messages that automatically inform them of events take place on the Liv-ex exchange (e.g. a trade taking place for one of their bids or offers). For more information please read the document "PUSH services".

**Base URI**

[exchange/v4/orders](https://exchange/v4/orders)

**Request Parameters**

| Name         | Mandatory | Description  |
|--------------|-----------|--|
| contractType | Y         | A valid contract type of the order. The possible value can be <b>sib</b> (Standard In Bond), <b>sep</b> (Standard En Primeur) and <b>x</b> (Special).<br><br><b>Type:</b> alphanumeric |

|                |                                      |  |
|----------------|--------------------------------------|--|
| special        | N<br>(mandatory if contractType = x) |  |
| dutyPaid       | N<br>(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).<br><b>Type:</b> Boolean 'true' or 'false'  |
| minimumQty     | N<br>(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.<br><b>Type:</b> integer   |
| deliveryPeriod | N<br>(mandatory if contractType = x) | States whether the lead time on the offer is different to the standard Liv-ex terms of 2 weeks.<br>If deliveryPeriod = 0, the offer is <b>Special Now</b> i.e. the stock is in the Liv-ex warehouse, has been checked and is ready for immediate dispatch.<br><b>Type:</b> integer |
| condition      | N<br>(mandatory if contractType = x) | A free text field that states any issues with the stock or its packaging.<br><b>Type:</b> string   |
| orderType      | Y                                    | A valid type of the order. The possible values can be <b>B</b> (bid) and <b>O</b> (offer).<br><b>Type:</b> alphanumeric  |
| orderStatus    | Y                                    | A valid order status. The possible values can be L (live) and S (suspended). See recommendation note.<br><b>Type:</b> alphanumeric   |
| expiryDate     | N                                    | A valid future date in yyyy-mm-dd format e.g. 2015-07-31.<br><b>Type:</b> alphanumeric, ISO8601 format   |
| lwin           | Y                                    | A valid LWIN7 or LWIN18 code<br><b>Type:</b> integer, 7-digits (LWIN7), 18-digits (LWIN18)   |
| vintage        | N<br>(mandatory if LWIN7 provided)   | Mandatory if LWIN7 is provided.<br>The value can be one year less than current year. For non-vintage wines use 1000.<br><b>Type:</b> 4-digit integer   |

|                  |  |   |
|------------------|--|---|
| bottleInCase     | N<br>(mandatory if LWIN7 provided)                       | Mandatory if LWIN7 is provided. Values are typically 1, 3, 6, 12, 24<br><b>Type:</b> 2-digit integer  |
| bottleSize       | N<br>(mandatory if LWIN7 provided)                       | Mandatory if LWIN7 is provided. The values must be in ml (millilitres) expressed as a 5-digit term e.g. 75cl = 00750.<br><b>Type:</b> 5-digit integer                                       |
| currency         | Y  | Either EUR or GBP. The currency used must match the trading currency pre-agreed with Liv-ex.<br><b>Type:</b> 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.<br><b>Type:</b> integer or double                             |
| quantity         | Y  | A valid positive integer value of quantity of packs such as 1,2, 50 etc.<br><b>Type:</b> integer  |
| 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.<br><b>Type:</b> alphanumeric (30-character limit) |
| specialOrderGUID | N<br>Mandatory if placing order against existing special | The order identification code of the Special position (bid or offer) that the new order should be placed against.<br><b>Type:</b> 128-bit hexadecimal                                       |

**Recommendation Notes:**

1. A successful POST will return an orderGUID value. This is the unique identifier used within the Liv-ex system for that order. A record of each orderGUID should be made as they are used to edit (PATCH) and delete (DELETE) orders. PUSH updates (if consumed) use orderGUID to identify specific events.
2. specialOrderGUID should be supplied when placing an order against a specific Special (contractType = x) that already exists on the exchange. The new order will inherit the contract terms of the Special position supplied.

**Sample Request Body**

The Orders service permits multiple orders to be supplied as an array/list. Clients may prefer to pass only one at a time. Examples of both are shown below.

### JSON Request (single order)

```
{
  "orders": [
    {
      "specialOrderGUID": "",
      "contractType": "SIB",
      "orderType": "o",
      "orderStatus": "L",
      "expiryDate": "2018-12-01",
      "lwin": "1006045",
      "vintage": "2012",
      "bottleInCase": "12",
      "bottleSize": "00750",
      "currency": "GBP",
      "price": "3400",
      "quantity": "1",
      "merchantRef": "PO #123456"
    }
  ]
}
```

### JSON Request (multiple orders)

```
{
  "orders": [
    {
      "specialOrderGUID": "",
      "contractType": "X",
      "special": {
        "dutyPaid": true,
        "minimumQty": "4",
        "deliveryPeriod": 2,
        "condition": "test"
      },
      "orderType": "o",
      "orderStatus": "L",
      "expiryDate": "2018-12-01",
      "lwin": "1006045",
      "vintage": "2012",
      "bottleInCase": "12",
      "bottleSize": "00750",
      "currency": "GBP",
      "price": "3200",
      "quantity": "4",
      "merchantRef": "place special offer using API"
    },
    {
      "specialOrderGUID": "",
      "contractType": "SIB",
      "special": {
        "dutyPaid": false,
        "minimumQty": "1",
        "deliveryPeriod": 2,
        "condition": ""
      },
      "orderType": "o",
      "orderStatus": "L",
      "expiryDate": "2018-12-01",
      "lwin": "1006045",
      "vintage": "2012",
      "bottleInCase": "06",
      "bottleSize": "00750",
      "currency": "GBP",
      "price": "1700",
      "quantity": "1",
    }
  ]
}
```



```
"merchantRef": "place SIB offer with all POST attributes "
}}
}
```

### XML Request (single order)

```
<Orders>
  <Order>
    <specialOrderGUID></specialOrderGUID>
    <contractType>sib</contractType>
    <orderType>o</orderType>
    <orderStatus>L</orderStatus>
    <expiryDate>2018-09-28</expiryDate>
    <lwin>1006045</lwin>
    <bottleInCase>12</bottleInCase>
    <bottleSize>00750</bottleSize>
    <currency>gbp</currency>
    <price>3400</price>
    <quantity>1</quantity>
    <merchantRef>PO #123456</merchantRef>
  </Order>
</Orders>
```

### XML Request (multiple orders)

```
<Orders>
  <Order>
    <specialOrderGUID></specialOrderGUID>
    <contractType>X</contractType>
    <special>
      <dutyPaid>>true</dutyPaid>
      <minimumQty>4</minimumQty>
      <deliveryPeriod>2</deliveryPeriod>
      <condition>test</condition>
    </special>
    <orderType>o</orderType>
    <orderStatus>L</orderStatus>
    <expiryDate>2018-09-28</expiryDate>
    <lwin>1006045</lwin>
    <bottleInCase>12</bottleInCase>
    <bottleSize>00750</bottleSize>
    <currency>gbp</currency>
    <price>3300</price>
    <quantity>4</quantity>
    <merchantRef>Placing special offer using API</merchantRef>
  </Order>
  <Order>
    <specialOrderGUID></specialOrderGUID>
    <contractType>X</contractType>
    <special>
      <dutyPaid>>false</dutyPaid>
      <minimumQty>1</minimumQty>
      <deliveryPeriod>2</deliveryPeriod>
      <condition></condition>
    </special>
    <orderType>o</orderType>
    <orderStatus>L</orderStatus>
    <expiryDate>2018-09-28</expiryDate>
    <lwin>1006045</lwin>
    <bottleInCase>06</bottleInCase>
```

```
<bottleSize>00750</bottleSize>
<currency>gbp</currency>
<price>1700</price>
<quantity>1</quantity>
<merchantRef>lace SIB offer with all POST attributes</merchantRef>
</Order>
</Orders>
```

### Sample Response Body

| 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.  |

### JSON Response

The response is sent per order

```
{
  "status": "OK",
  "httpCode": "200",
  "message": "Request completed successfully",
  "internalErrorCode": "R001",
  "apiInfo": {
    "version": "2.0",
    "timestamp": "1521454694828",
    "provider": "Liv-ex"
  },
  "orders": [{
    "merchantRef": "MyRef",
    "orderGUID": "94B5CC70-BC3D-49C3-B636-C3C7552E543D",
    "orderPlaceDate": "1521454694537"
  }],
  "errors": ""
}
```

### Invalid JSON response

```
{
  "status": "failure",
  "statusCode": "400",
  "internalErrorCode": "R002",
  "message": "Request partially completed",
  "apiInfo": {
    "version": "2.0",
    "timestamp": "1521454694828",
    "provider": "Liv-ex"
  },
  "orders": [{
    "merchantRef": "MyRef",
    "orderGUID": "",
    "orderPlaceDate": ""
    "errors": {
      "error": [{
        "code": "TR005",
        "message": "Please provide positive numeric value of qty."
      },
      {
        "code": "TR006",
        "message": "Please provide positive decimal value of price."
      }
    ]
  }
  ],
}
```

### XML Response

#### The response is sent per order

```
<exchangeResponse>
  <Status>OK</Status>
  <HttpCode>200</Code>
  <Message>Request completed successfully.</Message>
  <internalErrorCode>R001</internalErrorCode>
  <Orders>
    <Order>
      <merchantRef>MyRef</MerchantRef>
      <orderGUID>94B5CC70-BC3D-49C3-B636-C3C7552E543D</OrderId>
      <orderPlaceDate>2015-06-04T07:22:25</OrderPlaceDate>
      <Errors/>
    </Order>
  </Orders>
</exchangeResponse>
```

### Invalid XML Response

```

<exchangeResponse>
  <Status>failure</Status>
  <HttpCode>400</Code>
  <Message>Sorry, problem encountered while processing the Order.</Message>
  <internalErrorCode>R001</internalErrorCode>
  <ApiInfo>
    <Version>2.0</Version>
    <Timestamp>2015-06-04T11:12:30</Timestamp>
    <Provider>Liv-ex</Provider>
  </ApiInfo>
  <Orders>
    <Order>
      <merchantRef>MyRef</MerchantRef>
      <orderGUID/>
      <orderPlaceDate/>
      <Errors>
        <Error>
          <Code>TR005</Code>
          <Message>Please provide positive numeric value of qty.</Message>
        </Error>
        <Error>
          <Code>TR006</Code>
          <Message>Please provide positive decimal value of price</Message>
        </Error>
      </Errors>
    </Order>
  </Orders>
</exchangeResponse>

```

## 5.2 Edit Order Service (PATCH method)

### Description

This web service will be used to amend the bid or offer of a merchant.

The PATCH method allows for one or more attributes to be updated. Only the attribute(s) to be updated need to be supplied in the request (alongside the ID (orderGUID) of the resource to be modified).

Note: orderGUID is the resource identifier number returned when using the POST (add) order service.

### Base URI

[exchange/v4/orders](https://exchange/v4/orders)

### Parameters

| Name           | Mandatory                            | Description  |
|----------------|--------------------------------------|--|
| orderGUID      | Y                                    | The identifier of the order resource to be updated.<br><b>Type:</b> 128-bit hexadecimal  |
| contractType   | Y                                    | A valid contract type of the order. The possible value can be <b>sib</b> (Standard In Bond), <b>sep</b> (Standard En Primeur) and <b>x</b> (Special).<br><b>Type:</b> alphanumeric   |
| special        | N<br>(mandatory if contractType = x) | Required for Special (x) contract request structure. Has no attribute value.   |
| dutyPaid       | N<br>(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).<br><b>Type:</b> Boolean 'true' or 'false'  |
| minimumQty     | N<br>(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.<br><b>Type:</b> integer   |
| deliveryPeriod | N<br>(mandatory if contractType = x) | States whether the lead time on the offer is different to the standard Liv-ex terms of 2 weeks.<br>If deliveryPeriod = 0, the offer is <b>Special Now</b> i.e. the stock is in the Liv-ex warehouse, has been checked and is ready for immediate dispatch.<br><b>Type:</b> integer |

|             |                                      |   |
|-------------|--------------------------------------|---|
| condition   | N<br>(mandatory if contractType = x) | A free text field that states any issues with the stock or its packaging.<br><b>Type:</b> string  |
| orderStatus | Y                                    | A valid order status. The possible values can be L (live) and S (suspended). See recommendation note.<br><b>Type:</b> alphanumeric  |
| expiryDate  | N                                    | A valid future date in yyyy-mm-dd format e.g. 2015-07-31.<br><b>Type:</b> alphanumeric, ISO8601 format  |
| 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.<br><b>Type:</b> integer or double                             |
| quantity    | Y                                    | A valid positive integer value of quantity of packs such as 1,2, 50 etc.<br><b>Type:</b> integer  |
| 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.<br><b>Type:</b> alphanumeric (30-character limit) |

### Sample PATCH request body

|  |
|--|
| <p><b>JSON</b></p> <pre>{   "orders": [     {       "orderGUID": "f2f80a78-de1a-45b7-a20f-08faa19f2417",       "price": 3550,       "quantity": 7,       "merchantRef": "editing offer using PATCH method"     }   ] }</pre>   |
| <p><b>XML</b></p> <pre>&lt;Orders&gt;   &lt;Order&gt;     &lt;orderGUID&gt;d76a7329-9ba9-4e59-8658-f93c8f6ed7f2&lt;/orderGUID&gt;     &lt;merchantRef&gt;editing offer using PATCH method&lt;/merchantRef&gt;     &lt;orderStatus&gt;L&lt;/orderStatus&gt;     &lt;price&gt;1120&lt;/price&gt;     &lt;quantity&gt;2&lt;/quantity&gt;     &lt;expiryDate&gt;2018-09-28&lt;/expiryDate&gt;   &lt;/Order&gt; &lt;/Orders&gt;</pre> |

### JSON Response

**Response with valid orderGUID**

```
{
  "status": "OK",
  "statusCode": "200",
  "message": "Request completed successfully.",
  "internalErrorCode": "R001",
  "apiInfo": {
    "version": "4.0",
    "timestamp": 1537441620753,
    "provider": "Liv-ex"
  },
  "orders": {
    "order": [
      {
        "merchantRef": "editing offer using PATCH method",
        "orderGUID": "f2f80a78-de1a-45b7-a20f-08faa19f2417",
        "orderPlaceDate": 1537441620619,
        "errors": null
      }
    ]
  }
}
```

**Response with invalid orderGUID**

```
{
  "status": "Bad Request",
  "statusCode": "400",
  "message": "Request was unsuccessful.",
  "internalErrorCode": "R000",
  "apiInfo": {
    "version": "4.0",
    "timestamp": 1537441379582,
    "provider": "Liv-ex"
  },
  "orders": {
    "order": [
      {
        "merchantRef": null,
        "orderGUID": "f2f80a78-de1a-45b7-a20f-08faa19f2418",
        "orderPlaceDate": null,
        "errors": {
          "error": [
            {
              "code": "V056",
              "message": "GUID is not available or does not exist"
            }
          ]
        }
      }
    ]
  }
}
```

**XML Response**

**Response with valid orderGUID**

```
<?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>4.0</Version>
    <Timestamp>2018-09-20T12:17:43.696+01:00</Timestamp>
    <Provider>Liv-ex</Provider>
  </ApiInfo>
  <Orders>
    <order>
      <MerchantRef>Placing offer from API</MerchantRef>
      <OrderGUID>f2f80a78-de1a-45b7-a20f-08faa19f2417</OrderGUID>
      <OrderPlaceDate>2018-09-20T12:17:43.505+01:00</OrderPlaceDate>
      <Errors xsi:nil="true"/>
    </order>
  </Orders>
</exchangeResponse>
```

#### Response with invalid orderGUID

```
<?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>Bad Request</Status>
  <HttpCode>400</HttpCode>
  <Message>Request was unsuccessful.</Message>
  <InternalErrorCode>R000</InternalErrorCode>
  <ApiInfo>
    <Version>4.0</Version>
    <Timestamp>2018-09-20T12:19:05.522+01:00</Timestamp>
    <Provider>Liv-ex</Provider>
  </ApiInfo>
  <Orders>
    <order>
      <MerchantRef xsi:nil="true"/>
      <OrderGUID>f2f80a78-de1a-45b7-a20f-08faa19f2418</OrderGUID>
      <OrderPlaceDate xsi:nil="true"/>
      <Errors>
        <error>
          <code>V056</code>
          <message>GUID is not available or does not exist</message>
        </error>
      </Errors>
    </order>
  </Orders>
</exchangeResponse>
```



### 5.3 Delete Order Service (DELETE method)

#### Description

This web service will be used to delete the bid or offer of a merchant.

#### Base URI

[exchange/v4/orders](#)

#### Parameters

| Name      | Mandatory | Description                         |
|-----------|-----------|-------------------------------------|
| orderGUID | Y         | A valid order GUID for the account. |

#### Sample JSON DELETE request body

Order ID is a GUID number returned when using add order service

|  |
|--|
| <p><b>JSON</b></p> <pre>{   "orderGUID": ["94B5CC70-BC3D-49C3-B636-C3C7552E543D"] }</pre> <p><b>XML</b></p> <pre>&lt;orderDeleteRequest&gt;   &lt;orderGUID&gt;94B5CC70-BC3D-49C3-B636-C3C7552E543D &lt;/orderGUID&gt; &lt;/orderDeleteRequest&gt;</pre> |
|--|

#### JSON Response

##### Response with valid order ID

|   |
|---|
| <pre>{   "status": "OK",   "httpCode": "200",   "message": "Request completed successfully.",   "internalErrorCode": "R001",   "apiInfo": {     "version": "2.0",     "timestamp": "1462450114401",     "provider": "Liv-ex"   },   "orders": {     "order": {       "merchantRef": "MyRef",       "orderGUID": "94B5CC70-BC3D-49C3-B636-C3C7552E543D",       "orderPlaceDate": "1462450114337",       "errors": null     }   } }</pre> |
|---|

##### Response with invalid order ID

```
{
  "status": "failure",
  "httpCode": "400",
  "message": "Sorry, problem encountered while processing the Order.",
  "internalErrorCode": "R000",
  "apiInfo": {
    "version": "2.0",
    "timestamp": "1462450114401",
    "provider": "Liv-ex"
  },
  "orders": [
    {
      "merchantRef": "MyRef",
      "orderGUID": "94B5CC70-BC3D-49C3-B636-C3C7552E543D",
      "response": ""
    }
  ],
  "errors": [
    {
      "error": {
        "code": "TR001",
        "message": "Merchant and order combination does not match."
      }
    }
  ]
}
```

## XML Response

### Response with valid order ID

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<exchangeResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="https://api.liv-ex.com/v1 https://api.liv-ex.com/schema/v1/services.xsd">
  <Status>OK</Status>
  <HttpCode>200</HttpCode>
  <Message>Request completed successfully.</Message>
  <InternalErrorCode>R001</InternalErrorCode>
  <ApiInfo>
    <Version>2.0</Version>
    <Timestamp>2017-11-21T12:26:34.917Z</Timestamp>
    <Provider>Liv-ex</Provider>
  </ApiInfo>
  <Orders>
    <order>
      <MerchantRef>MyRef</MerchantRef>
      <OrderGUID>c1c9a11f-5c8f-4746-99d9-06bb4d2216b6</OrderGUID>
      <OrderPlaceDate>2017-11-21T12:26:34.838Z</OrderPlaceDate>
      <Errors xsi:nil="true"/>
    </order>
  </Orders>
</exchangeResponse>
```

### Response with invalid order ID

```

<Response>
  <Status>failure</Status>
  <HttpCode>400</Code>
  <internalErrorCode>R000</internalErrorCode>
  <Message>Request unsuccessful</Message>
  <ApiInfo>
    <Version>2.0</Version>
    <Timestamp>2017-11-04T11:12:30</Timestamp>
    <Provider>Liv-ex</Provider>
  </ApiInfo>
  <Orders>
    <Order>
      <merchantRef>MyRef</MerchantRef>
      <orderGUID>94B5CC70-BC3D-49C3-B636-C3C7552E543D</OrderId>
      <Errors>
        <Error>
          <Code>TR001</Code>
          <Message>Merchant and order combination does not match.</Message>
        </Error>
      </Errors>
    </Order>
  </Orders>
</Response>

```

## 6. Exchange validations override

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  
Bid or offers that are significantly higher or 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 manager if you would like to know more.

## 7. Supplementary API services

heartbeat API – check the exchange is available

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

orderByUID API – sell stock you have stored in the Liv-ex Vine warehouse as Special Now

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   |

|             |   |
|-------------|---|
| <b>V055</b> | Order details do not match orderGUID  |
| <b>V056</b> | orderGUID is not available or does not exist  |
| <b>V072</b> | Quantity change is not allowed in this order.   |
| <b>V073</b> | Invalid / incorrect dutyPaid: [<value>]. Possible values are 'true', 'false'  |
| <b>V074</b> | Invalid / incorrect deliveryPeriod: [<value>]. Must be positive integer value. Max value = 16 weeks   |
| <b>V075</b> | Invalid / incorrect minimumQty: [<value>]. Must be a positive integer value   |
| <b>V077</b> | Invalid / incorrect contractType: [<value>]. Possible values can be 'sib' (Standard In Bond), 'sep' (Standard En Primeur) and 'x' (Special).    |
| <b>V078</b> | Missing attribute: [<value>]. When orderType = 'o' and contractType = 'x', attributes dutyPaid, minimumQty and deliveryPeriod must be included. |
| <b>V085</b> | Parent and child orders can't have same order type.   |
| <b>V086</b> | Please provide valid special terms of contract to create a special order  |
| <b>V087</b> | Contract type change is not allowed in this order.  |

## 8.2 Trade validation error codes

| <b>Code</b>  | <b>Error Key</b>  | <b>Meaning</b>  |
|--------------|---|---|
| <b>TR001</b> | invalid.made.avail                                      | The deliver period supplied is not valid. Must be a positive integer value (max value = 16) |
| <b>TR002</b> | invalid.min.unit.and.qty                                | Your bid does not meet the minimum quantity terms of the contract                           |
| <b>TR003</b> | bid.fat.finger.weak.warn                                | A bid appears to be above Market Price. Please check carefully before proceeding.           |
| <b>TR004</b> | bid.fat.finger.strong.warn                              | A bid appears to be above Market Price. Please check carefully before proceeding.           |
| <b>TR005</b> | offer.fat.finger.weak.warn                              | An offer appears to be below Market Price. Please check carefully before proceeding.        |
| <b>TR006</b> | offer.fat.finger.strong.warn                            | An offer appears to be below Market Price. Please check carefully before proceeding.        |
| <b>TR007</b> | order.did.not.confirm.<br>successfully.try.again        | The order did not confirm successfully. Please check request syntax and try again.          |
| <b>TR010</b> | phy.offer.restrict.due.to.<br>exceeding.qty.than.stored | Merchant trading status is set to 'buy and resell only'.                                    |
| <b>TR011</b> | merchant.about.to.match.his.<br>offer                   | Merchant is about to match their own offer  |

|              |   |  |
|--------------|---|--|
| <b>TR012</b> | merchant.about.to.match.his.bid                     | Merchant is about to match their own bid   |
| <b>TR013</b> | merchant.awrs.urn.not.found                         | Merchant is placing a duty paid offer but does not have a validated AWRS number                |
| <b>TR014</b> | merchant.status.no.trading                          | Merchant does not have trading privileges.   |
| <b>TR015</b> | merchant.status.sell.only.no.phy.bid                | Merchant account is 'sell only'.   |
| <b>TR016</b> | merchant.status.sell.only.ep.resell.for.a.vintage   | Merchant is not permitted to buy EP stock. EP sell status is 'resell only'.                    |
| <b>TR017</b> | merchant.status.sell.only.no.ep.for.a.vintage       | Merchant is not permitted to sell EP stock.  |
| <b>TR018</b> | merchant.status.sell.only.ep.allowed                | Merchant is not permitted to sell EP stock.  |
| <b>TR019</b> | 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'. |
| <b>TR020</b> | sso.offer.restrict.due.to.exceeding.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:

| <b>Code</b>      | <b>Message</b>   |
|------------------|--|
| 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).<br><b>Type:</b> 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.<br><b>Type:</b> integer   |
| deliveryPeriod    | States whether the lead time on the offer is different to the standard Liv-ex terms of 2 weeks.<br>If deliveryPeriod = 0, the offer is <b>Special Now</b> i.e. the stock is in the Liv-ex warehouse, has been checked and is ready for immediate dispatch.<br><b>Type:</b> integer |

|           |  |
|-----------|--|
| condition | A free text field that states any issues with the stock or its packaging.<br><b>Type:</b> 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.

**Important – feed inconsistencies**

The **dutyPaid** and **condition** attributes are handled differently in the Exceloffer2 and Excelbid2 datafeeds. New, revised versions of the feeds will make them consistent with other Liv-ex services.

**dutyPaid**            stock is in bond = 'true'; stock is duty paid = 'dutypaid'

**condition**        no condition issue text = 'true'; condition issues = '<free text string>'