



Trading Activity API v2

Document Revision 1.0
Date of Issue: 2 April 2019
Date of revision: 2 April 2019

Daria Ershova

Table of Contents

1. Purpose	3
2. Glossary of Terms	3
3. Technical Standards	3
4. Request Header	3
5. API Listing	5
5.1 Trading Activity service (POST method).....	5
6. Response Codes	12
6.1 Request validation error codes.....	12
6.2 HTTP Status codes.....	13
7. Appendices	14
7.1 Appendix 1 – Time frame scenarios	14
7.2 Appendix 2 – Sorting order	Error! Bookmark not defined.

1. Purpose

To provide the API end point information and examples of the web services available for Trading Activity.

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).
Trade	A bid and offer match for a trade to take place on the Exchange for a certain amount of wine.
Contract Type	Contract type is a generic term for SIB, SEP or X (Special).
SIB	Standard in Bond trade terms: https://www.liv-ex.com/knowledge/liv-ex-trading-contracts/
SEP	Standard En Primeur: https://www.liv-ex.com/knowledge/liv-ex-trading-contracts/
X (Special)	Special contract trade terms: https://www.liv-ex.com/knowledge/liv-ex-trading-contracts/

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 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.

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. 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",
  "statusCode": "401",
  "message": "Request was unsuccessful",
  "livexCode": "R000"
  "apiInfo": {
    "version": "2.0",
    "timestamp": 1551628884,
    "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>2019-03-03T11:12:30</Timestamp>
    <Provider>Liv-ex</Provider>
  </ApiInfo>
</Response>
```

5. API Listing

5.1 Trading Activity service (POST method)

Description

This service will retrieve the status and invoice details of the merchant's Liv-ex trades.

The only mandatory request attribute is 'statusType'. Calling the service with just this attribute using 'invoiced' and 'pending' values will result in all trades with this statusType being returned; for 'completed' data only the last 30 days of information will be returned by default (longer time ranges are also available). The remaining non-mandatory attributes can be used to sort, and filter results as required.

Base URI

[accounts/v2/tradingActivity](#)

Request Parameters

Name	Mandatory	Description
statusType	Y	Values: 'invoiced', 'pending', 'completed' Type: alphanumeric
sortingOrder	N Default = 'status'	Values: 'date', 'status' Type: alphanumeric <i>Only one value permitted per request</i>
completedTimeRange	N Applicable only to 'completed' trades. Default = '30'	Values: '30', '90', '180', '1year', '2year' Type: alphanumeric <i>Only one value permitted per request</i>
lxTradeNo	N	Liv-ex number of the wine trade, e.g. "207165", "lx207165", "LX207165", "Lx207165" Type: alphanumeric <i>Only one value permitted per request</i>
merchantRef	N	Query by reference text. Max amount of characters is 30. Type: alphanumeric <i>Only one value permitted per request</i>

Sample Request Body

JSON Request

```
{
  "tradingActivity": {
    "statusType": [ "pending", "invoiced"],
    "sortingOrder": "date",
    "completedTimeRange": "",
    "lxTradeNo": "",
    "merchantRef": ""
  }
}
```

XML Request

```
<root>
<tradingActivity>
  <statusType>invoiced</statusType>
  <statusType>pending</statusType>
  <sortingOrder>date</sortingOrder>
  <completedTimeRange></completedTimeRange>
  <lXTradeNo></lXTradeNo>
  <merchantRef></merchantRef>
</tradingActivity>
</root>
```

Sample Response Body

The Positions service will respond with HTTP Code 200 OK in a successful response.

Response parameters

Name	Description
lXTradeNo	Type: integer Example: 208973
lXTradeDate	Type: Alphanumeric, ISO 8601 Epoch time if JSON Example (JSON): 1479372182898 Example (XML): 2015-06-04T11:12:30
statusType	Type: alphanumeric Example: invoiced, pending, completed
role	Type: alphanumeric Example: Buy, Sell
lwin	Type: integer Example: 123456720080600750
wineName	Type: alphanumeric Example: Figeac
vintage	Type: 4-digit integer Example: 2014, 1000 (for NV)
packSize	Type: 2-digit integer Example: 6
bottleSize	Type: 5-digit integer Example: 00750
contractType	Type: alphanumeric Example: SIB, SEP, X (for special)
quantity	Type: integer Example: 1
unitPrice	Type: double Example 500
merchantRef	Type: alphanumeric Example: 116400
status	Type: alphanumeric Example: In Account – Awaiting payment
invoiceDate	Type: alphanumeric, ISO 8601 Epoch time if JSON

	Example (JSON): 1479372182898 Example (XML): 2015-06-04T11:12:30
expectedPaymentDate	Type: alphanumeric, ISO 8601 Epoch time if JSON Example (JSON): 1479372182898 Example (XML): 2015-06-04T11:12:30
invoiceNo	Type: integer Example: 259827
currency	Type: alphanumeric Example: GBP
amount	Type: double Example: 512.25
vat	Type: double Example: 2.45
total	Type: double Example: 514.70
dateCompleted	Type: Alphanumeric, ISO 8601 Epoch time if JSON <i>Null unless statusType = completed</i> Example (JSON): 1479372182898 Example (XML): 2015-06-04T11:12:30
isPrepaid	Type: Boolean true/false <i>Applicable only to trades where Role = Buy.</i>

JSON Response

The response is sent per request.

```
"status": "OK",
  "statusCode": "200",
  "message": "Request completed successfully.",
  "internalErrorCode": "R001",
  "apiInfo": {
    "version": "2.0",
    "timestamp": 1554208846234,
    "provider": "Liv-ex"
  },
  "tradingActivities": [
    {
      "lxTradeNo": "208354",
      "lxTradeDate": 1539099389000,
      "statusType": "invoiced",
      "role": "Buy",
      "lwin": "100170520070600750",
      "wineName": "Clarendon Hills, Piggott Range Syrah",
      "vintage": 2007,
      "packSize": 6,
      "bottleSize": "00750",
      "contractType": "X",
      "quantity": 1,
      "unitPrice": 150,
      "merchantRef": "1000027074-150558",
      "status": "Released - Awaiting payment",
      "invoiceDate": 1539952086000,
      "expectedPaymentDate": 1559905595000,
      "invoiceNo": 275388,
```

```

    "currency": "GBP",
    "amount": 156.5,
    "vat": 1.3,
    "total": 157.8,
    "dateCompleted": null
    "isPrepaid": false
  },
  {
    "lxTradeNo": "209761",
    "lxTradeDate": 1538937678000,
    "statusType": "invoiced",
    "role": "Buy",
    "lwin": "100765120101200750",
    "wineName": "Cantemerle",
    "vintage": 2010,
    "packSize": 12,
    "bottleSize": "00750",
    "contractType": "SIB",
    "quantity": 1,
    "unitPrice": 285,
    "merchantRef": "2000048127-150403",
    "status": "In Account - Awaiting payment",
    "invoiceDate": 1539951973000,
    "expectedPaymentDate": 1559905595000,
    "invoiceNo": 249338,
    "currency": "GBP",
    "amount": 294.2,
    "vat": 1.84,
    "total": 296.04,
    "dateCompleted": null,
    "isPrepaid": false
  },
  {
    "lxTradeNo": "209173",
    "lxTradeDate": 1539245404000,
    "statusType": "pending",
    "role": "Sell",
    "lwin": "110203720150600750",
    "wineName": "Sassicaia",
    "vintage": 2015,
    "packSize": 6,
    "bottleSize": "00750",
    "contractType": "SIB",
    "quantity": 1,
    "unitPrice": 725,
    "merchantRef": "50000",
    "status": "Please deliver or or before 08/11/2018",
    "invoiceDate": null,
    "expectedPaymentDate": null,
    "invoiceNo": null,
    "currency": "GBP",
    "amount": 743,
    "vat": 3.6,
    "total": 746.6,
    "dateCompleted": null,
    "isPrepaid": null
  },
  {
    "lxTradeNo": "199900",
    "lxTradeDate": 1524239026000,
    "statusType": "invoiced",
    "role": "Buy",
    "lwin": "100609020160600750",
    "wineName": "Armailhac",
    "vintage": 2016,
    "packSize": 6,

```



```

        "bottleSize": "00750",
        "contractType": "SEP",
        "quantity": 1,
        "unitPrice": 192,
        "merchantRef": "11776",
        "status": "EP - Due In June 2019",
        "invoiceDate": 1525437146000,
        "expectedPaymentDate": 1559905595000,
        "invoiceNo": 2000061,
        "currency": "GBP",
        "amount": 199.34,
        "vat": 1.47,
        "total": 200.81,
        "dateCompleted": null,
        "isPrepaid": false
    }
  ],
  "errors": null
}

```

Invalid JSON response

```

{
  "status": "Bad Request",
  "statusCode": "400",
  "message": "Request was unsuccessful.",
  "internalErrorCode": "R000",
  "apiInfo": {
    "version": "2.0",
    "timestamp": 1554209896991,
    "provider": "Liv-ex"
  },
  "tradingActivities": null,
  "errors": {
    "error": [
      {
        "code": "V097",
        "message": "Invalid / incorrect statusType: [pending, invoice]. Possible values are 'invoiced', 'pending' or 'completed'."
      },
      {
        "code": "V098",
        "message": "Invalid / incorrect sortOrder: [dates]. Possible values are 'status' or 'date'."
      }
    ]
  }
}

```

XML Response

The response is sent per request.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<tradingActivitiesResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="https://aby-uat-api.liv-ex.com/v1 https://aby-uat-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>2019-04-02T13:48:44.823+01:00</Timestamp>
    <Provider>Liv-ex</Provider>
  </ApiInfo>
  <tradingActivities>

```

```

<lxTradeNo>208354</lxTradeNo>
<lxTradeDate>2018-10-09T16:36:29.000+01:00</lxTradeDate>
<statusType>invoiced</statusType>
<role>Buy</role>
<lwin>100170520070600750</lwin>
<wineName>Clarendon Hills, Piggott Range Syrah</wineName>
<vintage>2007</vintage>
<packSize>6</packSize>
<bottleSize>00750</bottleSize>
<contractType>X</contractType>
<quantity>1</quantity>
<unitPrice>150.0</unitPrice>
<merchantRef>1000027074-150558</merchantRef>
<status>Released - Awaiting payment</status>
<invoiceDate>2018-10-19T13:28:06.000+01:00</invoiceDate>
<expectedPaymentDate>2018-10-26T13:28:06.000+01:00</expectedPaymentDate>
<invoiceNo>275388</invoiceNo>
<currency>GBP</currency>
<amount>156.5</amount>
<vat>1.3</vat>
<total>157.8</total>
<dateCompleted xsi:nil="true"/>
</tradingActivities>
<tradingActivities>
  <lxTradeNo>209761</lxTradeNo>
  <lxTradeDate>2018-10-07T19:41:18.000+01:00</lxTradeDate>
  <statusType>invoiced</statusType>
  <role>Buy</role>
  <lwin>100765120101200750</lwin>
  <wineName>Cantemerle</wineName>
  <vintage>2010</vintage>
  <packSize>12</packSize>
  <bottleSize>00750</bottleSize>
  <contractType>SIB</contractType>
  <quantity>1</quantity>
  <unitPrice>285.0</unitPrice>
  <merchantRef>2000048127-150403</merchantRef>
  <status>In Account - Awaiting payment</status>
  <invoiceDate>2018-10-19T13:26:13.000+01:00</invoiceDate>
  <expectedPaymentDate>2018-10-26T13:28:06.000+01:00</expectedPaymentDate>
  <invoiceNo>249338</invoiceNo>
  <currency>GBP</currency>
  <amount>294.2</amount>
  <vat>1.84</vat>
  <total>296.04</total>
  <dateCompleted xsi:nil="true"/>
  <isPrepaid>false</isPrepaid>
</tradingActivities>
<tradingActivities>
  <lxTradeNo>209173</lxTradeNo>
  <lxTradeDate>2018-10-11T09:10:04.000+01:00</lxTradeDate>
  <statusType>pending</statusType>
  <role>Sell</role>
  <lwin>110203720150600750</lwin>
  <wineName>Sassicaia</wineName>
  <vintage>2015</vintage>
  <packSize>6</packSize>
  <bottleSize>00750</bottleSize>
  <contractType>SIB</contractType>
  <quantity>1</quantity>
  <unitPrice>725.0</unitPrice>
  <merchantRef>50000</merchantRef>
  <status>Please deliver or or before 08/11/2018</status>
  <invoiceDate xsi:nil="true"/>
  <expectedPaymentDate xsi:nil="true"/>
  <invoiceNo xsi:nil="true"/>

```

```

<currency>GBP</currency>
<amount>743.0</amount>
<vat>3.6</vat>
<total>746.6</total>
<dateCompleted xsi:nil="true"/>
<isPrepaid xsi:nil="true"/>
</tradingActivities>
<tradingActivities>
  <lxTradeNo>199900</lxTradeNo>
  <lxTradeDate>2018-04-20T16:43:46.000+01:00</lxTradeDate>
  <statusType>invoiced</statusType>
  <role>Buy</role>
  <lwin>100609020160600750</lwin>
  <wineName>Armailhac</wineName>
  <vintage>2016</vintage>
  <packSize>6</packSize>
  <bottleSize>00750</bottleSize>
  <contractType>SEP</contractType>
  <quantity>1</quantity>
  <unitPrice>192.0</unitPrice>
  <merchantRef>11776</merchantRef>
  <status>EP - Due In June 2019</status>
  <invoiceDate>2018-05-04T13:32:26.000+01:00</invoiceDate>
  <expectedPaymentDate>2018-10-26T13:28:06.000+01:00</expectedPaymentDate>
  <invoiceNo>2000061</invoiceNo>
  <currency>GBP</currency>
  <amount>199.34</amount>
  <vat>1.47</vat>
  <total>200.81</total>
  <dateCompleted xsi:nil="true"/>
  <isPrepaid>false</isPrepaid>
</tradingActivities>
</tradingActivitiesResponse>

```

Invalid XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<tradingActivitiesResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="https://aby-uat-api.liv-ex.com/v1 https://aby-uat-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>2.0</Version>
    <Timestamp>2019-04-02T13:58:47.395+01:00</Timestamp>
    <Provider>Liv-ex</Provider>
  </ApiInfo>
  <errors>
    <error>
      <code>V097</code>
      <message>Invalid / incorrect statusType: [invoice, pending]. Possible values
are 'invoiced', 'pending' or 'completed'.</message>
    </error>
    <error>
      <code>V098</code>
      <message>Invalid / incorrect sortingOrder: [dates]. Possible values are
'status' or 'date'.</message>
    </error>
  </errors>
</tradingActivitiesResponse>
```

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
V097	Invalid / incorrect statusType: [value]. Possible values are 'invoiced', 'pending' or 'completed'.
V098	Invalid / incorrect sortingOrder: [value]. Possible values are 'status' or 'date'.
V099	Invalid / incorrect completedTimeRange: [value]. Possible values are '30', '90', '180', '1year' or '2year'.
V100	"Liv-Ex trade number: [value] is not available or does not exist."
V101	Liv-Ex trade number: [value] does not exist for this combination of statusType.

V102	Merchant reference: [value] does not exist.
V103	For completed trades older than two years please contact Liv-Ex.
V104	No records found for this combination of statusType.
V105	Merchant reference: [value] does not correspond to the Liv-Ex trade number.

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.

7. Appendices

7.1 Appendix 1 – Time frame scenarios

statusType	completedTimeRange	Result
Invoiced	Null	All records
Invoiced	'30', '90', '180', '1year', '2 year'	All records
Pending	Null	All records
Pending	'30', '90', '180', '1year', '2 year'	All records
Completed	Null	Records for the last 30 days
Completed	'30', '90', '180', '1year', '2 year'	As per selected time frame (e.g. 2 years)
Invoiced, completed	Null	All records for invoiced, records for the last 30 days for completed trades
Invoiced, completed	'30', '90', '180', '1year', '2 year'	All records for invoiced, records for the time frame requested for completed trades
Pending, completed	Null	All records for pending, records for the last 30 days for completed trades
Pending, completed	'30', '90', '180', '1year', '2 year'	All records for pending, records for the time frame requested for completed trades
Invoiced, pending	Null	All records
Invoiced, pending	'30', '90', '180', '1year', '2 year'	All records
Completed, invoiced, pending	Null	All records for invoiced and pending, records for the last 30 days for completed trades
Completed, invoiced, pending	'30', '90', '180', '1year', '2 year'	All records for invoiced and pending, records for the time frame requested for completed trades