Price Data API Documentation

Document Revision 1.2
Date of Issue: 04 November 2016
Date of revision: 09 December 2016

Nick Palmer

Business Systems Analyst

Table of Contents

1. Purpose

To provide the API end-point information and examples of the data available from the Price Data API.

2. Glossary of Terms

Term	Meaning
LWIN	LWIN - the Liv-ex Wine Identification Number – serves as a universal wine identifier for the wine trade. A unique seven to eighteen-digit numerical code 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 cheapest 6 and 12-pack prices advertised by leading merchants in the EU. (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.
Best Bid	The highest price at which Liv-ex trading members are currently bidding to buy the wine on the market.
Best Offer	The lowest price at which Liv-ex trading members are currently offering to sell the wine on the market.
Mid Price	Used to calculate the Liv-ex indices and value the world's leading fine wine funds. The mid-point between the highest live bid and lowest live offer on the market, validated against additional data including transaction prices.
Auction Price	The 30-day average hammer price collected from all major auction houses.
Last Trade	The most recent Liv-ex trade price.
Offex	Transaction data gathered from members who provide information on trades conducted through their own sales channels. The data is anonymised and turned into a 30-day-average price.
Average List Price	The 30-day mean price for all stock listed by Liv-ex merchants.
Listed Quantity	Number of 9 litre cases available from Liv-Ex members base on lists supplied to us over the last 30 days. An aggregate of 6 and 12 bottle cases.
MID Last List Price	The last list price supplied to us by the Merchant who is using the Price Data API service.
MID Last Offex Price	The last Offex price supplied to us by the Merchant who is using the Price Data API service.

Contract Type	Contract type is a generic term for SIB, SEP or Special (X).
SIB	Standard in Bond trade terms (see Liv-ex rules & regulations)
SEP	Standard En Primeur (see Liv-ex rules and regulations)
X	Special contract terms (see Liv-ex rules and regulations)

3. Technical Standards

- Permitted users will be issued with a unique token (CLIENT_KEY) and password (CLIENT_SECRET) combination to control the access to the web service.
- 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. GET for read operation
    3. PUT for update operation
    4. DELETE for delete operation
- 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 PUT & DELETE
- Authentication mechanism will be custom based on CLIENT_KEY and CLIENT_SECRET
- For PUSH services we require a direct POST URL which should be backed by a service capable of accepting and process XML payload as POST request.
- 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 must provide the following information in the request header.

Parameter

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	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.

Header example

```
CLIENT_KEY:     94B5CC70-BC3D-49C3-B636-C3C7552E543D
CLIENT_SECRET:  merchantpasswd
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": "2015-06-04T11:12:30",
       "provider": "Liv-ex"
   }
}
```

Invalid header XML response

```
<Response>
      <Status>Unauthorized</Status>
      <HttpCode>401</Code>
      <Message>Request was unsuccessful</Message>

      <LivexCode>R001</LivexCode>

                <ApiInfo>
       <Version>1.0</Version>
       <Timestamp>2015-06-04T11:12:30</Timestamp>
       <Provider>Liv-ex</Provider>
     </ApiInfo>

<Response>
```

5. API Listing

5.1 Price Data (POST Method)

Description

This service will be used to retrieve a range of Liv-Ex data points for a given LWIN code.

Base URI

/data/v1/priceData

Request Parameters

<table>
<tr><th>Name</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>lwin</td><td>Y</td><td>LWIN7/11/16/18. LWINs must all be of the same length in each request.

Type: alphanumeric

Example JSON: “lwin”: [“0123456”,”1234567”]
Example XML: <lwin>0123456</lwin>
<lwin>1234567</lwin></td></tr>
<tr><td>priceType</td><td>Y</td><td>Defines the price data type to return. Multiple price types are permitted per request and using either the codes or alternative codes listed below:

<table>
<tr><th>Price Data type</th><th>code</th><th>alt. code</th></tr>
<tr><td>Market price</td><td>A</td><td>marketPrice</td></tr>
<tr><td>Best bid</td><td>B</td><td>bestBid</td></tr>
<tr><td>Best offer</td><td>C</td><td>bestOffer</td></tr>
<tr><td>Mid price</td><td>D</td><td>midPrice</td></tr>
<tr><td>Auction price</td><td>E</td><td>auctionPrice</td></tr>
<tr><td>Trade Price</td><td>F</td><td>lastTrade</td></tr>
<tr><td>Offex</td><td>G</td><td>offexPrice</td></tr>
<tr><td>Average list price</td><td>H</td><td>averageListPrice</td></tr>
<tr><td>Listed quantity</td><td>I</td><td>averageListQty</td></tr>
<tr><td>Your last list price</td><td>J</td><td>yourLastList</td></tr>
<tr><td>Your last Offex</td><td>K</td><td>yourLastOffex</td></tr>
</table>

Type: alphanumeric

Example JSON: “priceType”: [”A”,”bestOffer”]
Example XML: <priceType>A</priceType>
<priceType>BestOffer</priceType></td></tr>
<tr><td>currency</td><td>Y</td><td>Desired response currency. Only one currency may be specified in each request. The following</td></tr>
</table>

		currencies are available: GBP, USD, JPY, EUR, CHF, SGD, HKD. **Type**: 3-character alphanumeric **Example JSON:** "currency": "GBP" **Example XML:** <currency>GBP</currency>
priceDate	N	A date field in the format yyyy-mm-dd that can be any day in the past. If left blank will default to current date. **Type:** alphanumeric, ISO8601 format **Example JSON:** "priceDate": "2016-01-20" **Example XML:** <priceDate>2016-01-20</priceDate>
vintage	N	Optional for L-WIN7 requests. The value can be one year less than current year. For non-vintage use 1000. **Type**: 4-digit integer **Example JSON:** "vintage": "2004" **Example XML:** <vintage>2004</vintage>
bottleSize	N	Optional for LWIN7 or LWIN11 requests. The values must be in ml (millilitres). **Type**: 5-digit integer **Example JSON:** "bottleSize": "00750" **Example XML:** <bottleSize>00750</bottleSize>
packSize	N	Optional for LWIN7, LWIN11 and LWIN16 requests. **Type**: 2-digit integer **Example JSON:** "packSize": "12" **Example XML:** <packSize>12</packSize>

Requests must only contain LWINs of one length (i.e. all LWIN16s or all LWIN18s) Multiple LWINs can be passed in the form of an array. A maximum of 50 LWINs can be included in each request.

Note: Pricing data is limited for LWIN7 requests unless a vintage is also supplied (forming an LWIN11). Current (live) data is limited to 4 of the 11 price types. No historic data is available for LWIN7.

Price Data type	**code**	**alt. code**	**LWIN7 response**
Market price	*A*	*marketPrice*	The vintage of that wine witht the lowest market price
Best bid	*B*	*bestBid*	The vintage of that wine with the lowest live bid price
Best offer	*C*	*bestOffer*	The vintage of that wine the highest live offer price
Trade Price	*F*	*lastTrade*	The vintage of that wine with the lowest recent trade price.

Validation code V043 wil be returned if a dataType unavailable for LWIN7 is requested.

Sample JSON Request Body (POST method)

GBP best offer for Pontet Canet 2010 12x75

```
{
    "lwin": ["101430720101200750"],
    "priceType": ["B"],
    "priceDate": "",
    "currency": "GBP"

}
```

Average List Price, Market Price and Last Trade for Pontet Canet 2010

```
{
    "lwin": ["101430720101200750"],
    "priceType": ["H","A","F"],
    "priceDate": "",
    "currency": "GBP"

}
```

Sample XML Request Body (POST method)

GBP best offer for Pontet Canet 2010 12x75

```
<priceDataRequest>
    <lwin>101430720101200750</lwin>
    <priceType>bestOffer</priceType>
    <currency>GBP</currency>
</priceDataRequest>
```

Average List Price, Market Price and Last Trade for Pontet Canet 2010

```
<priceDataRequest>
    <lwin>10143072010</lwin>
    <priceType>averageListPrice</priceType>
    <priceType>marketPrice</priceType>
    <priceType>lastTrade</priceType>
    <currency>GBP</currency>
</priceDataRequest>
```

Response Parameters

The API returns the associated data point to the LWIN code(s) supplied:

- LWIN7 requests return data for the cheapest vintage of that wine. Vintages may vary between price data types. Not all priceTypes are available for LWIN7 requests.
- LWIN11 requests return data one the cheapest 9l equivalent price.
- LWIN16 requests return price data specific to the bottle format requested.
- LWIN18 requests return price data specific to pack and bottle format requested.

Parameter Name	Mandatory	Description

lwin	Y	LWIN7/11/16/18 code requested. **Type:** alphanumeric **Example:** 012345620080600750
iwp	Y	URL link to the IWP page of the LWIN returned. **Type**: alphanumeric **Example**: http://www.liv-ex.com/individualWine.do?vinWinId=64846
priceType	Y	The type of price data being returned. Multiple price types may be returned in each response permitted per request: **Type:** alphanumeric **Example:** A
priceData	Y	The requested pricing data. Note that trailing zeros will not be returned. **Type**: Numeric to 2dp. **Example**: 5150.25
priceQty	N	The associated quantity of stock available on the exchange, specific to the pack &bottle size shown in the response. **Type**: numeric (integer) **Example**: 4
priceDate	Y	The date and time (where applicable) when the priceData requested was updated by Liv-Ex. See table 1 (below) for more details. **Type**: Alphanumeric, ISO 8601 **Example (JSON):** 1479372182898 **Example (XML):** 2015-06-04T11:12:30
currency	Y	The currency of the data point(s) returned **Type**: 3-character alphanumeric **Example**: GBP
vintage	Y	The vintage of the wine corresponding to the data returned. **Type**: 4-digit integer **Example**: 2004
bottleSize	Y	The bottle size corresponding to the data returned in ml (millilitres). **Type**: 5-digit integer

		Example: 00750
packSize	Y	The pack size corresponding to the data returned. **Type**: 2-digit integer **Example**: 12
contractType	N	Best Bid / Best Offer only (priceType B and C). Return the contract type associated with the bid or offer. **Type**: alphanumeric **Example**: SIB
otherPositions	N	Best Bid / Best Offer only (priceType B and C). Return a value of true if there are additional bid/offers on alternative pack sizes. **Type**: Boolean **Example**: 1

priceData type	**Code**	**alt. code**	**priceDate response: current data**	**priceDate response: historic data**
Market price	A	marketPrice	date	date
Best bid	B	bestBid	date & time	date
Best offer	C	bestOffer	date & time	date
Mid price	D	midPrice	date	date
Auction price	E	auctionPrice	date	date
Last trade	F	lastTrade	date & time	date & time
Offex	G	offexPrice	date	date
Average list price	H	averageListPrice	date	date
Listed quantity	I	averageListQty	date	date
Your last list price	J	yourLastList	date	date
Your last Offex price	K	yourLastOffex	Date	date

Table 1 - priceDate responses

JSON response – success

GBP best offer for Pontet Canet 2010 12x75


```
{
  "status": "OK",
  "httpCode": "200",
  "message": "Request completed successfully.",
  "internalErrorCode": "R001",
  "apiInfo": {
    "version": "1.0",
```

```
    "timestamp": 1479372182898,
    "provider": "Liv-ex"
  },
  "lwinDetail": [
    {
      "lwin": "101430720101200750",
      "iwp": "http://54.194.231.39:8091/individualWine.do?vinWinId=52061",
      "dataDetail": [
        {
          "priceType": "B",
          "priceData": "1687.0",
          "priceDate": "2016-10-03",
          "priceQty": 1,
          "currency": "GBP",
          "vintage": "2010",
          "packSize": "12",
          "bottleSize": "00750",
          "contractType": "X",
          "otherPositions": "Y",
          "error": null
        }
      ],
      "error": null
    }
  ],
  "errors": null
}
```


Average List Price, Market Price and Last Trade for Pontet Canet 2010


```
{
  "status": "OK",
  "httpCode": "200",
  "message": "Request completed successfully.",
  "internalErrorCode": "R001",
  "apiInfo": {
    "version": "1.0",
    "timestamp": 1479372182898,
    "provider": "Liv-ex"
  },
  "lwinDetail": [
    {
      "lwin": "101430720101200750",
      "iwp": "http://54.194.231.39:8091/individualWine.do?vinWinId=52061",
      "dataDetail": [
        {
          "priceType": "A",
          "priceData": "1598.0",
          "priceDate": "2016-10-03",
          "priceQty": 1,
          "currency": "GBP",
          "vintage": "2010",
          "packSize": "12",
          "bottleSize": null,
          "contractType": null,
          "otherPositions": null,
          "error": null
        }
```

```
      ],
      "dataDetail": [
        {
          "priceType": "H",
          "priceData": "1600.0",
          "priceDate": "2016-09-30",
          "priceQty": 1,
          "currency": "GBP",
          "vintage": "2010",
          "packSize": "12",
          "bottleSize": "00750",
          "contractType": null,
          "otherPositions": null,
          "error": null
        }
      ],
      "dataDetail": [
        {
          "priceType": "F",
          "priceData": "1932.0",
          "priceDate": "2016-11-30",
          "priceQty": null,
          "currency": "GBP",
          "vintage": "2010",
          "packSize": "12",
          "bottleSize": "00750",
          "contractType": null,
          "otherPositions": null,
          "error": null
        }
      ],

      "error": null
    }
  ],
  "errors": null
}
```


JSON response - failure


```
{
    "status": "Unauthorized",
    "httpCode": "401",
    "message": "Unauthorized",
    "internalErrorCode": null,
    "apiInfo":
    {
        "version": "1.0",
        "timestamp": 1478265178795,
        "provider": "Liv-ex"
    }
}
```


XML response – success

GBP best offer for Pontet Canet 2010 12x75

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<priceDataResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="https://api.liv-ex.com/v1
http://54.154.139.15:8091/schema/v1/services.xsd">
    <Status>OK</Status>
    <HttpCode>200</HttpCode>
    <Message>Request completed successfully.</Message>
    <InternalErrorCode>R001</InternalErrorCode>
    <ApiInfo>
        <Version>1.0</Version>
        <Timestamp>2016-11-17T08:51:36.194Z</Timestamp>
        <Provider>Liv-ex</Provider>
    </ApiInfo>
    <lwinDetail>
        <lwin>10143072010120750</lwin>
        <iwp>http://54.194.231.39:8091/individualWine.do?vinWinId=121610</iwp>
        <dataDetail>
            <priceType>bestOffer</priceType>
            <priceData>1687.0</priceData>
            <priceDate>2016-10-05T00:00:00.000+01:00</priceDate>
            <priceQty>1</priceQty>
            <currency>GBP</currency>
            <vintage>2010</vintage>
            <packSize>12</packSize>
            <bottleSize>00750</bottleSize>
            <contractType>X</contractType>
            <otherPositions>Y</otherPositions>
            <error xsi:nil="true"/>
        </dataDetail>
        <error xsi:nil="true"/>
    </lwinDetail>
    <errors xsi:nil="true"/>
</priceDataResponse>
```

Average List Price, Market Price and Last Trade for Pontet Canet 2010

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<priceDataResponse 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>1.0</Version>
        <Timestamp>2016-11-30T14:33:19.847Z</Timestamp>
        <Provider>Liv-ex</Provider>
    </ApiInfo>
    <lwinDetail>
        <lwin>10143072010</lwin>
        <iwp>http://54.194.231.39:8091/individualWine.do?vinWinId=58262</iwp>
        <dataDetail>
            <priceType>marketPrice</priceType>
            <priceData>1598.0</priceData>
            <priceDate>2016-10-05T00:00:00.000+01:00</priceDate>
            <priceQty>1</priceQty>
            <currency>GBP</currency>
           <vintage>2010</vintage>
```

```
            <packSize>12</packSize>
            <bottleSize>00750</bottleSize>
            <contractType xsi:nil="true"/>
            <otherPositions xsi:nil="true"/>
            <error xsi:nil="true"/>
        </dataDetail>
        <dataDetail>
            <priceType>lastTrade</priceType>
            <priceData>1600.0</priceData>
            <priceDate>2016-09-30T09:26:39.000+01:00</priceDate>
            <priceQty>1</priceQty>
            <currency>GBP</currency>
            <vintage>2010</vintage>
            <packSize>12</packSize>
            <bottleSize>00750</bottleSize>
            <contractType xsi:nil="true"/>
            <otherPositions xsi:nil="true"/>
            <error xsi:nil="true"/>
        </dataDetail>
        <dataDetail>
            <priceType>averageListPrice</priceType>
            <priceData>1932.0</priceData>
            <priceDate>2016-11-30T00:00:00.000Z</priceDate>
            <priceQty xsi:nil="true"/>
            <currency>GBP</currency>
            <vintage>2010</vintage>
            <packSize>12</packSize>
            <bottleSize>00750</bottleSize>
            <contractType xsi:nil="true"/>
            <otherPositions xsi:nil="true"/>
            <error xsi:nil="true"/>
        </dataDetail>
        <error xsi:nil="true"/>
    </lwinDetail>
    <errors xsi:nil="true"/>
</priceDataResponse>
```

XML response – failure

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Response xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="https://api.liv-ex.com/v1
http://54.154.139.15:8091/schema/v1/services.xsd">
    <Status>Unauthorized</Status>
    <HttpCode>401</HttpCode>
    <Message>Unauthorized</Message>
    <InternalErrorCode xsi:nil="true"/>
    <ApiInfo>
        <Version>1.0</Version>
        <Timestamp>2016-11-17T08:53:20.730Z</Timestamp>
        <Provider>Liv-ex</Provider>
    </ApiInfo>
</Response>
```

6. Response Codes

This section describes the response codes that will be returned by the Price Data API service.

Code	Message
R000	Request was unsuccessful
R001	Request completed successfully
R002	Request partially completed

6.1 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 L-WIN number.
V012	Invalid request headers. Please provide value for {header name}.
V013	Please provide valid vintage.
V015	Invalid currency.
V038	Requested priceType () does not exist
V039	You do not have permission to access (priceType). Please contact Liv-Ex.
V040	Wrong historic date format. Date should be 'yyyy-MM-dd'.
V041	Requested date should be a valid date within the last () years(s).
V042	API limited to a maximum of 50 LWIN codes per request.
V043	Data is not available for LWIN7 requests

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 or invalid authentication details are provided. Also useful to trigger an auth popup if the 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.
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.