



## Price Data API v2

Document Revision 2.5  
Date of Issue: 08 March 2019  
Date of revision: 27 July 2022

Daria Ershova  
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>4</b>
<b>4. Request Header</b> .....	<b>4</b>
<b>5. API Listing</b> .....	<b>6</b>
5.1 Price Data (POST Method) .....	6
<b>6. Response Codes</b> .....	<b>15</b>
6.1 Request validation error codes .....	15
6.2 HTTP Status codes .....	15

## 1. Purpose

To provide the API end-point information and examples of the data available from the Price Data API v2.

## 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. 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.
<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 bids and offers.
<b>Market Price</b>	The cheapest list price advertised by leading merchants. It provides a guide as to the price you are likely to pay for SIB-compliant stock in the market.
<b>Best bid</b>	The highest price at which Liv-ex trading members are currently bidding to buy the wine on the market.
<b>Best offer</b>	The lowest price at which Liv-ex trading members are currently offering to sell the wine on the market.
<b>Mid price</b>	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.
<b>Auction price</b>	The 30-day average hammer price collected from all major auction houses.
<b>Last auction price</b>	Data from the most recent wine auction event to take place for that wine.
<b>Last trade price</b>	The most recent Liv-ex trade price.
<b>Offex price</b>	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.
<b>Average list price</b>	The 30-day mean price for all stock listed by Liv-ex merchants.
<b>Listed quantity</b>	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.
<b>Your last list price</b>	The last list price supplied to us by the Merchant who is using the Price Data API service.
<b>Your last offex price</b>	The last Offex price supplied to us by the Merchant who is using the Price Data API service.
<b>Ex-chateau release price</b>	The release price of the wine at the Chateau gate. Only relevant to wines sold En Primeur and only available to negociants. Always a EUR/bottle value
<b>Ex-negociant release price</b>	The release price of the wine from the negociant. Only relevant to wines sold En Primeur and only available to wine wholesalers and retailers. Always a EUR/bottle value
<b>Ex-London release price</b>	The release price of the wine by the major merchants in London. Only relevant to wines sold En Primeur. Always a GBP/case (9 litre) value

<b>Contract type</b>	Contract type describes the terms under which the order (bid or offer) is placed. Either SIB, SEP or Special (X).
<b>SIB</b>	Standard in Bond trade terms: <a href="https://www.liv-ex.com/knowledge/liv-ex-trading-contracts/">https://www.liv-ex.com/knowledge/liv-ex-trading-contracts/</a>
<b>SEP</b>	Standard En Primeur: <a href="https://www.liv-ex.com/knowledge/liv-ex-trading-contracts/">https://www.liv-ex.com/knowledge/liv-ex-trading-contracts/</a>
<b>X (Special)</b>	Special contract trade terms: <a href="https://www.liv-ex.com/knowledge/liv-ex-trading-contracts/">https://www.liv-ex.com/knowledge/liv-ex-trading-contracts/</a>

### 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 will support the following methods:
  1. POST for read 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 or JSON 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 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 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 or 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 or 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",
  "statusCode": "401",
  "message": "Request was unsuccessful",
  "livexCode": "R000"
  "apiInfo": {
    "version": "2.0",
    "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>2.0</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 variety of Liv-ex data points for a given LWIN code.

#### Base URI

</data/v2/priceData>

#### Request Parameters

Name	Mandatory	Description																																																
lwin	Y	LWIN7/11/16/18. Multiple values are permitted (max. 50 per request). <u>LWINs must all be of the same length per request.</u> <b>Type:</b> string array																																																
priceType	Y	Defines the price data type to return. Multiple values are permitted. Request payloads may use either the attribute names or alternative codes listed below: <table border="1" data-bbox="867 982 1409 1661"> <thead> <tr> <th>Price data type</th> <th>attribute name</th> <th>alt. code</th> </tr> </thead> <tbody> <tr> <td>Market price</td> <td>marketPrice</td> <td>A</td> </tr> <tr> <td>Best bid</td> <td>bestBid</td> <td>B</td> </tr> <tr> <td>Best offer</td> <td>bestOffer</td> <td>C</td> </tr> <tr> <td>Mid price</td> <td>midPrice</td> <td>D</td> </tr> <tr> <td>Auction price</td> <td>auctionPrice</td> <td>E</td> </tr> <tr> <td>Trade Price</td> <td>lastTrade</td> <td>F</td> </tr> <tr> <td>Offex</td> <td>offexPrice</td> <td>G</td> </tr> <tr> <td>Average list price</td> <td>averageListPrice</td> <td>H</td> </tr> <tr> <td>Listed quantity</td> <td>averageListQty</td> <td>I</td> </tr> <tr> <td>Your last list price</td> <td>yourLastList</td> <td>J</td> </tr> <tr> <td>Your last Offex</td> <td>yourLastOffex</td> <td>K</td> </tr> <tr> <td>Last auction price</td> <td>lastAuction</td> <td>L</td> </tr> <tr> <td>Ex-chateau price</td> <td>exChateau</td> <td>M</td> </tr> <tr> <td>Ex-negociant price</td> <td>exNegociant</td> <td>N</td> </tr> <tr> <td>Ex-London price</td> <td>exLondon</td> <td>O</td> </tr> </tbody> </table> <b>Type:</b> string array	Price data type	attribute name	alt. code	Market price	marketPrice	A	Best bid	bestBid	B	Best offer	bestOffer	C	Mid price	midPrice	D	Auction price	auctionPrice	E	Trade Price	lastTrade	F	Offex	offexPrice	G	Average list price	averageListPrice	H	Listed quantity	averageListQty	I	Your last list price	yourLastList	J	Your last Offex	yourLastOffex	K	Last auction price	lastAuction	L	Ex-chateau price	exChateau	M	Ex-negociant price	exNegociant	N	Ex-London price	exLondon	O
Price data type	attribute name	alt. code																																																
Market price	marketPrice	A																																																
Best bid	bestBid	B																																																
Best offer	bestOffer	C																																																
Mid price	midPrice	D																																																
Auction price	auctionPrice	E																																																
Trade Price	lastTrade	F																																																
Offex	offexPrice	G																																																
Average list price	averageListPrice	H																																																
Listed quantity	averageListQty	I																																																
Your last list price	yourLastList	J																																																
Your last Offex	yourLastOffex	K																																																
Last auction price	lastAuction	L																																																
Ex-chateau price	exChateau	M																																																
Ex-negociant price	exNegociant	N																																																
Ex-London price	exLondon	O																																																
currency	Y	The desired response currency. Only one value is permitted per request. <b>Values:</b> 'gbp', 'eur', 'chf', 'usd', 'hkd', 'jpy', 'sgd', 'gbp/btt', 'eur/btt', 'chf/btt', 'usd/btt', 'hkd/btt', 'jpy/btt', 'sgd/btt'																																																

		<b>Type:</b> alphanumeric
priceDate	N	Used for historic data calls. Can be any date in the past but range of data available may be restricted by Liv-ex. If left blank will default to current date <b>Type:</b> alphanumeric, ISO8601 format (yyyy-mm-dd)
vintage	N	<i>Optional for L-WIN7 requests</i> The value can be one year less than current year. For non-vintage use 1000. <b>Type:</b> 4-digit integer
bottleSize	N	<i>Optional for LWIN7 or LWIN11 requests</i> The values must be in ml (millilitres) using a 5-digit format. <b>Type:</b> 5-digit integer Example: 750ml bottle = 00750
packSize	N	<i>Optional for LWIN7, LWIN11 and LWIN16 requests</i> <b>Type:</b> 2-digit integer Example: 6-bottle pack = 06

**Operational notes:**

1. A request must only contain LWINs of one length (i.e. all LWIN16s or all LWIN18s) or it will be rejected. Please use multiple requests if querying at a variety of LWIN lengths. A maximum of 50 LWINs can be included in each request.
2. Price data values are returned differently depending on the LWIN length submitted.

LWIN length	Data response
LWIN7	Considers all vintages available and returns information on the single vintage that holds the best/cheapest value.
LWIN11	Considers all pack and bottle format combination of the vintage specified by the LWIN11
LWIN16	Considers prices that correspond to the specific bottleSize format in the LWIN16
LWIN18	Considers prices that correspond to the specific bottleSize and packSize in the LWIN18

3. LWIN7 behaviour

- LWIN requests are limited to current marketPrice, bestbid, bestOffer and lastTrade data. For all other data types LWIN11 or higher is required. Historic data requests are not available for LWIN7s. Validation code V043 will be returned when a priceType not valid for LWIN7 is requested.
- LWIN7 requests for permitted priceTypes work as follows:

priceType	LWIN7 response
marketPrice	The vintage of the requested LWIN7 with the lowest current market price
bestBid	The vintage of the requested LWIN7 with the lowest current live bid price
bestOffer	The vintage of the requested LWIN7 with the highest current live offer price
lastTrade	The vintage of the requested LWIN7 with the lowest recent trade price

### 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 12x75

```
{
  "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

Parameter Name	Description
lwin	LWIN7/11/16/18 code requested. Type: alphanumeric Example: 012345620080600750
iwp	URL link to the IWP page of the LWIN returned. Type: alphanumeric Example: <a href="http://www.liv-ex.com/individualWine.do?vinWinId=64846">http://www.liv-ex.com/individualWine.do?vinWinId=64846</a>
priceType	The type of price data being returned. Multiple price types may be returned in each response permitted per request: Type: alphanumeric Example: A
priceData	The requested pricing data. Note that trailing zeros will not be returned. Type: Numeric to 2dp. Example: 5150.25
priceQty	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	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	The currency of the data point(s) returned Type: alphanumeric Example: GBP
vintage	The vintage of the wine corresponding to the data returned. Type: alphanumeric Example: 2004
bottleSize	The bottle size corresponding to the data returned in ml (millilitres). Type: alphanumeric Example: 00750
packSize	The pack size corresponding to the data returned. Type: alphanumeric Example: 12
contractType	Best Bid / Best Offer only (priceType B and C). Return the contract type associated with the bid or offer. Type: alphanumeric Example: SIB
metadata1	Additional information associated with the data point. Not always used. Type: alphanumeric
metadata2	Additional information associated with the data point. Not always used. Type: alphanumeric
otherPositions	Best Bid / Best Offer only (priceType B and C).

	<p>Return a value of true if there are additional bid/offers on alternative pack sizes. Type: Boolean true/false Example: 1</p>
--	---

**Operational notes:**

1. lastTrade priceData values are calculated from a normalised 9 litre value. The API returns a value in proportion to this data. Example: a 6x75 format request will return the 9 litre value divided by 2.
2. For lastTrade requests at the LWIN16 or LWIN18 level, the bottleSize and packSize data returned will always correspond to that made in the request. The price point will be pro-rata'd to the format configuration requested.
3. For historic lastTrade requests, the API will return the same priceData value as made in the request. Note that the last trade event this data relates to could have happened prior to this date.
4. priceDate content varies depending on the priceType requested and whether it is for current or historic data:

attribute name	alt. code	Price date format	
		Current data	Historic data
marketPrice	A	date	date
bestBid	B	date & time	date
bestOffer	C	date & time	date
midPrice	D	date	date
auctionPrice	E	date	date
lastTrade	F	date & time	date & time
offexPrice	G	date	date
averageListPrice	H	date	date
averageListQty	I	date	date
yourLastList	J	date	date
yourLastOffex	K	date	date
lastAuction	L	date	date
exChateau	M	date	date
exNegociant	N	date	date
exLondon	O	date	date

**JSON response – success**

**GBP best offer for Pontet Canet 2010 12x75**

```
{
  "status": "OK",
  "statusCode": "200",
  "message": "Request completed successfully.",
  "internalErrorCode": "R001",
  "apiInfo": {
    "version": "2.0",
    "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",
            "metadata1": null,
            "metadata2": null,
            "otherPositions": "Y",
            "error": null
          }
        ],
        "error": null
      }
    ],
    "errors": null
  }

```

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

```

{
  "status": "OK",
  "statusCode": "200",
  "message": "Request completed successfully.",
  "internalErrorCode": "R001",
  "apiInfo": {
    "version": "2.0",
    "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,
          "metadata1": null,
          "metadata2": 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,
    "metadata1": null,
    "metadata2": 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,
    "metadata1": null,
    "metadata2": null,
    "otherPositions": null,
    "error": null
  }
],
"error": null
}
],
"errors": null
}

```

**JSON response - failure**

```

{
  "status": "Unauthorized",
  "statusCode": "401",
  "message": "Unauthorized",
  "internalErrorCode": null,
  "apiInfo":
  {
    "version": "2.0",
    "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>2.0</Version>
    <Timestamp>2016-11-17T08:51:36.194Z</Timestamp>
    <Provider>Liv-ex</Provider>
  </ApiInfo>
  <lwinDetail>
    <lwin>101430720101200750</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>
      <metadata1/>
      <metadata2/>
      <otherPositions>Y</otherPositions>
      <error/>
    </dataDetail>
  </lwinDetail>
</errors/>
</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>2.0</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/>
      <metadata1/>
      <metadata2/>
    </dataDetail>
  </lwinDetail>
</errors/>
</priceDataResponse>
```

```

        <otherPositions/>
        <error/>
    </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/>
        <metadata1/>
        <metadata2/>
        <otherPositions/>
        <error/>
    </dataDetail>
    <dataDetail>
        <priceType>averageListPrice</priceType>
        <priceData>1932.0</priceData>
        <priceDate>2016-11-30T00:00:00.000Z</priceDate>
        <priceQty/>
        <currency>GBP</currency>
        <vintage>2010</vintage>
        <packSize>12</packSize>
        <bottleSize>00750</bottleSize>
        <contractType/>
        <metadata1/>
        <metadata2/>
        <otherPositions/>
        <error/>
    </dataDetail>
    <error/>
</lwinDetail>
<errors/>
</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/>
    <ApiInfo>
        <Version>2.0</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.