Pre-advice API Documentation

Document Revision 1.7
Date of Issue: 31 January 2017
Date of revision: 04 August 2020

Nick Palmer

Product Lead

Table of Contents

1. Purpose

To provide the API end-point information and examples of the data than can be sent via the Pre-advice API service.

2. API Behaviour

Important: the vTrans code assigned to each line of pre-advice is not 100% persistent.

It cannot be guaranteed that each individual case will be landed under the vTrans that corresponds with the case on the pre-advice. The reason being is that the pre-advice may refer to two wines of the same LWIN that in reality are different. For example, two cases of the same wine, one in perfect condition, one with a condition issue are sent to Vine. Upon arrival at the warehouse the vTrans and case UIDs are assigned, but the staff receiving the stock will not know which vTrans goes with which case. Only once the cases have been checked by the Passport team will the cases be opened and the condition issue identified.

The same scenario happens for two cases of the same wine with difference prices. The team physically receiving the case will not know which case is the cheaper one and will therefore assign a vTrans and UID in a random order.

To resolve this issue, users may wish to call the CellarView2 API to run a nightly reconciliation. The majority of pre-advices and stock checks will tie up. A small number may need checking or amending in your system to swap over any vTrans that might have been assigned and swapped between cases.

In summary, 99% of the time, the vtrans numbers will match between the pre-advice receipt and CellarView2 call. For the 1% (probably much less than that), a nightly reconciliation report should be run and any discrepancies flagged for manual checking / amending.

3. 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 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.
SIB	Standard in Bond trade terms: http://www.liv-ex.com/staticPageContent.do?pageKey=Rules_and_Regulations

SEP	Standard En Primeur: http://www.liv-ex.com/staticPageContent.do?pageKey=Rules_and_Regulations
Contract Type	Contract type is a generic term for SIB, SEP or Special (X).
Trade	A bid and offer match for a trade to take place on the Exchange for a certain amount of wine.
UID	UID is Liv-ex's unique identification number allocated to a case of wine in the Vine warehouse.
In Bond (IB)	Wines 'in bond' have not yet had the Duty and VAT paid on them. They must be stored in a bonded warehouse approved by HM Customs & Excise.
Duty Paid (DP)	Purchased wines which have passed through customs, with UK Duty and VAT paid on them.
Vtrans	The unique Vine transaction number associates with each pre-advice line.
Purchase Order number	A user provided purchase order number/reference associated with one or more items of stock.

4. 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

5. 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/ 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>
```

6. API Listing

6.1 Pre-advice (POST method)

Description

This service will be used to send pre-advice instructions to Liv-Ex.

Base URI

https://api.liv-ex.com/logistics/v1/preAdvice

Request Parameters

Name	Mandatory	Description
purchaseOrder	Y	The customer purchase order number associated with this line. **Type:** alphanumeric (30-character limit) **Example:** po12345
lwin	Y	LWIN7/11/16/18. LWINs must all be of the same length in each request. **Type:** numeric field of 7, 11, 16 or 18 chararcters **Example:** 123456720011200750
dutyStatus	Y	The tax status of the stock being advised **Type:** alphanumeric (2 characters) **Example:** DP
quantity	Y	The number of cases of the wine being pre-advised based on the supplied unitSize. **Type**: numeric **Example:** 7
unitPrice	Y	The price per unit of the wine. Must be a positive value and can include up to 2 decimal places. **Type:** double **Example:** 2015.75
currency	Y	The currency in which the pre-advice stock price is being supplied. **Type**: 3-character alphanumeric **Example:** GBP
supplier	Y	The name of your supplier. 50-character limit. **Type:** alphanumeric (50)

		Example: ABC Fine Wines
passportRequest	Y	Is an SIB Passport check required for the case(s) when it is received at Vine? Note: Lines with tax status duty paid ('DP') are not elegible for SIB Passport checks. DP lines requesting an SIB passport will return an error. **Type**: Boolean true/false **Example:** false
photoRequest	Y	Is a condition photo required for the case(s) when received at Vine? **Type**: Boolean true/false **Example:** true
subaccount	N	The Vine subaccount into which the wine should be placed. **Type:** alphanumeric **Example:** John Smith
vintage	N	Mandatory if LWIN7 supplied. For non-vintage use 1000. **Type:** 4-digit integer **Example:** 2004
bottleSize	N	Mandatory if LWIN7, LWIN11 supplied The value must be in ml (millilitres). **Type:** 5-digit integer **Example**: 00750
packSize	N	Mandatory if LWIN7, LWIN11 or LWIN16 supplied **Type:** 2-digit integer **Example:** 12

Usage Recommendation

To edit or update a pre-advice instruction, a delete request must be sent for the purchaseOrder or vTrans and then the line resent in a new POST request.

Sample JSON Request Body (POST method)

```
{"preAdvice":[
    {
    "lwin": "102346720001200750",
    "purchaseOrder": "123po123",
    "dutyStatus": "IB",
    "quantity": 10,
```

```
    "supplier": "merchant_1"
    "unitPrice": "950",
    "currency": "GBP",
    "passportRequest": true,
    "photoRequest": true,
    "subAccount": "ABC001"
    },
    {
    "lwin": "102346720010600750",
    "purchaseOrder": "123po123",
    "dutyStatus": "IB",
    "quantity": 4,
    "supplier": "merchant_1"
    "unitPrice": "500",
    "currency": "GBP",
    "passportRequest": true,
    "photoRequest": true,
    "subAccount": "DEF321"
    }
]}
```

Sample XML Request Body (POST method)

```
<preAdviceRequest>
  <preAdvice>
    <lwin>102345672000</lwin>
    <purchaseOrder>123po123</purchaseOrder>
    <dutyStatus>IB</dutyStatus>
    <quantity>10</quantity>
    <bottleSize>00750</bottleSize>
    <packSize>12</packSize>
    <supplier>merchant_1</supplier>
    <unitPrice>950</unitPrice>
    <currency>GBP</currency>
    <passportRequest>true</passportRequest>
    <photoRequest>true</photoRequest>
    <subaccount>ABC001</subaccount>
  </preAdvice>
  <preAdvice>
    <lwin>102345672001</lwin>
    <purchaseOrder>123po123</purchaseOrder>
    <dutyStatus>IB</dutyStatus>
    <quantity>10</quantity>
    <bottleSize>00750</bottleSize>
    <packSize>6</packSize>
    <supplier>merchant_1</supplier>
    <unitPrice>500</unitPrice>
    <currency>GBP</currency>
    <passportRequest>true</passportRequest>
    <photoRequest>true</photoRequest>
    <subaccount>DEF321</subaccount>
  </preAdvice>
</preAdviceRequest>
```

Response Parameters

The API response reports the status of the POST request followed by the status of each line of pre-advice contained within. Each successfully processed line will be assigned a unique Vtrans number. Lines that are rejected will be noted in the response and include an error code and message with the reason for the rejection.

Parameter Name	Mandatory	Description
lineNumber	Y	An integer value corresponding to each line of pre-advice sent in the POST request **Type:** Integer **Example:** 1
purchaseOrder	Y	The purchaseOrder submitted in the request. **Type:** alphanumeric (30-character limit) **Example:** po12345
vTrans	Y	The unique Vine transaction number associates with each pre-advice line. Can be used with DELETE method to delete specific stock lines. **Type:** Alphanumeric value beginning with 'V' **Example:** V123456
lwin	Y	The LWIN18 associated with the vTrans number. **Type:** 18-digiit integer **Example:** 123456720001200750

JSON response – success


```
{
  "status": "OK",
  "httpCode": "200",
  "message": "Request completed successfully.",
  "internalErrorCode": "R001",
  "apiInfo": {
    "version": "1.0",
    "timestamp": 1479372182898,
    "provider": "Liv-ex"
  },
  "preAdviceDetail": [
    {
      "status": "SUCCESS",
      "lineNumber": "1",
      "vTrans": "123456",
      "purchaseOrder": "test",
      "lwin": "102346720001200750",
      "error": null
    }
  ],
```

```
}
```

JSON response - failure

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

XML response - success

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<preAdviceResponse 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>2017-01-10T08:51:36.194Z</Timestamp>
        <Provider>Liv-ex</Provider>
    </ApiInfo>
    <preAdviceDetail>
        <errors xsi:nil="true"/>
        <lineNumber>1</lineNumber>
        <lwin>102346720001200750</lwin>
        <purchaseOrder>test</purchaseOrder>
        <status>SUCCESS</status>
        <vTrans>123456</vTrans>
    </preAdviceDetail>
</preAdviceResponse>
```

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>
```

6.2 Delete Pre-advice (DELETE method)

Description

This service will be used to delete pre-advice instructions sent to Liv-Ex.

Base URI

https://api.liv-ex.com/logistics/v1/preAdvice

Request Parameters

Name	Mandatory	Description
purchaseOrder	Y (if vTrans not provided)	The customer purchase order number associated with this line. The service will attempt to delete all lines associated with this purchase order reference. Items with stock movement status A or B (in account, booking in) cannot be deleted. **Type:** alphanumeric **Example:** po12345
vTrans	Y (if purchaseOrder not provided)	Vine transaction number The service will attempt to delete only the Vtrans specified. Items with stock movement status A or B (in account, booking in) cannot be deleted. **Type:** Alphanumeric. The leading 'V' of the vTrans code may be omitted. **Example:** V260755

Sample JSON Request Body (DELETE method)

```
{"preAdvice":[
    {
    "purchaseOrder": "po12345"
    }
]}
```

Sample XML Request Body (DELETE method)

```
<preAdviceRequest>
  <preAdvice>
    <vTrans>V260755</vTrans>
  </preAdvice>
</preAdviceRequest>
```

Response Parameters

The API response contains infomation on the status of the DELETE request. and on each of the lines of pre-advice it contains. Each successfully processed line will be assigned a unique Vtrans number. Lines that are rejectedwill be noted in the response alongside an error code and message with the reason why the line was not processed.

Parameter Name	Mandatory	Description
lineNumber	Y	An integer value corresponding to each line of pre-advice sent in the POST request **Type:** Integer **Example:** 1
purchaseOrder	Y	The purchaseOrder submitted in the request. **Type:** alphanumeric (30-character limit) **Example:** po12345
vTrans	Y	The unique Vine transaction number associates with each pre-advice line. Can be used with DELETE method to delete specific stock lines. **Type:** Alphanumeric value beginning with 'V' **Example:** V123456
lwin	Y	The LWIN18 associated with the vTrans number. **Type:** 18-digiit integer **Example:** 123456720001200750

JSON response – success

```
{
  "status": "OK",
  "httpCode": "200",
  "message": "Request completed successfully.",
  "internalErrorCode": "R001",
  "apiInfo": {
    "version": "1.0",
    "timestamp": 1479372182898,
    "provider": "Liv-ex"
  },
  "preAdviceDetail": [
    {
      "status": "SUCCESS",
      "lineNumber": "1",
      "purchaseOrder": null,
      "vTrans": "261044",
      "LWIN": "",
      "error": null
    }
  ],
  "errors": null
}
```


JSON response - failure


```
{
    "status": "Bad Request",
    "httpCode": "400",
    "message": "Request was unsuccessful. ",
    "internalErrorCode": "R000",
    "apiInfo":
    {
        "version": "1.0",
        "timestamp": 1478265178795,
        "provider": "Liv-ex"
    },
    "preAdviceDetail": [
      {
      "status": "ERROR",
}
```


XML response - success


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<preAdviceResponse 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>
    <preAdviceDetail>
        <purchaseOrder></purchaseOrder>
        <vTrans></vTrans>
```

```
        <LWIN></LWIN>
        <error xsi:nil="true"/>
    </preAdviceDetail>
    <errors xsi:nil="true"/>
</preAdviceResponse>
```

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>
```

7. 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

7.1 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.
V018	<line_no> Mandatory field missing (<missing_field>)
V044	SIB Passport cannot be requested for duty paid stock
V045	Please provide a valid bottle size
V046	Please provide a valid pack size
V047	Please provide purchase order or Vtrans reference
V048	Purchase order: <purchaseorder_number> does not exist
V049	Vtrans reference: <vtrans_number> does not exist
V050	API limited to a maximum of <#> lines per request
V051	Unable to delete Vtrans: <vtrans_number>, stock has been received

7.2 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)
207 Multiple Statuses	The message body contains several separate responses of differing statuses
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.