Indices List API

Document revision 1.0
Date of Issue: 28 July 2021
Date of revision: 28 July 2021

Barnabas Mullan

Business Analyst

Table of Contents

1. Purpose

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

2. Glossary of Terms

Term	Meaning
Liv-ex Index	Liv-ex has created several indices to track the price of fine wine, including the industry benchmark Liv-ex 100. The Liv-ex indices which track the prices of the most traded fine wines on the market, using the Liv-ex Mid Price.

3. Technical Standards

- Permitted users will be issued with a unique token (CLIENT_KEY) and password (CLIENT_SECRET) combination to control the access for all the web services covered under Exchange Integration.
- The web services will consume and produce both XML and JSON. The user can provide the content type in the request header. If the user does not provide any information, then the default content type will be JSON.
- The project will support ISO 8601.
- The project will only support HTTPS protocol for client and server communications.
- The API will support the following method(s):
    1. GET for read 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 Indices List API will be accessible at https://api.liv-ex.com/data/v1/indicesList

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.

Param

Name	Mandatory	Description
CLIENT_KEY	Y	A valid merchant GUID which will be unique for each merchant.
CLIENT_SECRET	Y	Password/Secret for the merchants CLIENT_KEY.
ACCEPT	N	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	N	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.

e.g.
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": "1509753600",
        "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>2017-11-04T11:12:30</Timestamp>
        <Provider>Liv-ex</Provider>
    </ApiInfo>
<Response>
```

5. API Listing

5.1 Indices list (GET)

Description

This may be used by merchant systems to retrieve a full list of Liv-ex indices with descriptions and their short names.

Base URI

data/v1/indicesList

Response

The API service will response with HTTP Code 200 OK in a successful response to the GET request with the valid credentials provided within the request header.

Response parameters

Name	Description
indexName	Name of the Liv-ex index Type: alphanumeric
indexShortName	Short name of the Liv-ex index, e.g. LX50 Type: alphanumeric
description	Descriptor for the Liv-ex index Type: alphanumeric
type	Denotes whether the components of the index are taken globally or regionally. Type: alphanumeric
subIndexOf	Notes the parent index of the specific Liv-ex Index Type: alphanumeric
bloombergName	The Bloomberg name for the Liv-ex Index Type: alphanumeric
thomsonReutersName	The Thomson Reuters name for the Liv-ex Index Type: alphanumeric
lastRebased	Date that the index was last given a new base level. Type: alphanumeric, ISO 8601 Epoch time if JSON Example (JSON): 1549537950898 Example (XML): 2019-02-07T11:12:30

Sample JSON Response


```
{
   "status": "OK",
   "httpCode": "200",
   "message": "Request completed successfully",
   "internalErrorCode": "R001",
   "apiInfo": {
      "version": "1.0",
      "timestamp": 1628245842415,
      "provider": "Liv-ex"
   },
   "indicesList": [
      {
         "indexName": "Liv-ex Fine Wine 50",
         "indexShortName": "LVX50",
         "description": "The Liv-
ex Fine Wine 50 Index tracks the daily price movement of the most heavily traded commodities in the fine wine market – the Bor
deaux First Growths. It includes only the ten most recent vintages; 2008-
2017 for most, and excludes En Primeur. No other qualifying criteria applied.",
         "type": "major",
         "subIndexOf": null,
         "bloombergName": "LIVXFW50",
         "thomsonReutersName": "LIVFW50",
         "lastRebased": null
```

```
    },
    {
      "indexName": "Liv-ex Fine Wine 100",
      "indexShortName": "LVX100",
      "description": "The Liv-
ex Fine Wine 100 Index is the industry leading benchmark. It represents the price movement of 100 of the most sought-
after fine wines on the secondary market",
      "type": "major",
      "subIndexOf": null,
      "bloombergName": "LIVX100",
      "thomsonReutersName": "LIVF100",
      "lastRebased": null
    },
    {
      "indexName": "Liv-ex Bordeaux 500",
      "indexShortName": "LVXBDX500",
      "description": "The Liv-ex Bordeaux 500 is Liv-
ex's most comprehensive index for Bordeaux wines. It represents the price movement of 500 leading wines from the region and i
s calculated monthly using the Liv-ex Mid Price. It has been backdated to December 2003. The index comprises six sub-
indices: the Fine Wine 50, the Right Bank 50, the Second Wine 50, the Sauternes 50, the Right Bank 100 and the Left Bank 200.",
      "type": "regional",
      "subIndexOf": "Liv-ex Fine Wine 1000",
      "bloombergName": null,
      "thomsonReutersName": null,
      "lastRebased": null
    }
  ],
  "errors": null
}
```


Sample XML Response


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<root>
  <Status>OK</Status>
  <HttpCode>200</HttpCode>
  <Message>Request completed successfully</Message>
  <InternalErrorCode>R001</InternalErrorCode>
  <ApiInfo>
    <Version>1.0</Version>
    <Timestamp>2021-08-06T10:31:57.638Z</Timestamp>
    <Provider>Liv-ex</Provider>
  </ApiInfo>
  <indicesList>
    <indexName>Liv-ex Fine Wine 50</indexName>
    <indexShortName>LVX50</indexShortName>
    <description>The Liv-
ex Fine Wine 50 Index tracks the daily price movement of the most heavily traded commodities in the fine wine market – the Bor
deaux First Growths. It includes only the ten most recent vintages; 2008-
2017 for most, and excludes En Primeur. No other qualifying criteria applied.</description>
    <type>major</type>
    <subIndexOf xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
```

```
    <bloombergName>LIVXFW50</bloombergName>
    <thomsonReutersName>LIVFW50</thomsonReutersName>
    <lastRebased xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
  </indicesList>
  <indicesList>
    <indexName>Liv-ex Fine Wine 100</indexName>
    <indexShortName>LVX100</indexShortName>
    <description>The Liv-
ex Fine Wine 100 Index is the industry leading benchmark. It represents the price movement of 100 of the most sought-
after fine wines on the secondary market</description>
    <type>major</type>
    <subIndexOf xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
    <bloombergName>LIVX100</bloombergName>
    <thomsonReutersName>LIVF100</thomsonReutersName>
    <lastRebased xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
  </indicesList>
  <errors xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
</root>
```


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 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.