



Invoices API v1

Document Revision 1.0
Date of Issue: 4 April 2019
Date of revision: 18 December 2020

Table of Contents

1. Purpose	3
2. Glossary of Terms	3
3. Technical Standards	3
4. Request Header	3
5. API Listing	4
5.1 Invoices service (POST method)	4
6. Response Codes	11
6.1 Request validation error codes	11
6.2 HTTP Status codes	12

1. Purpose

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

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.

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

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.

		<p>The values for the content type will be application/json or application/xml.</p> <p>If no/ invalid content type is found in the request, then JSON format will be used by default.</p>
--	--	---

Example header

<pre>CLIENT_KEY: 12A34BC56-DE7F-89G0-H1J2345K678L CLIENT_SECRET: dummy_password ACCEPT: application/json CONTENT-TYPE: application/json</pre>

Invalid header (JSON response)

<pre>{ "status": "Unauthorized", "httpCode": "401", "message": "Unauthorized", "internalErrorCode": null, "apiInfo": { "version": "1.0", "timestamp": 1554364615297, "provider": "Liv-ex" } }</pre>

Invalid header (XML response)

<pre><Response 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>Unauthorized</Status> <HttpCode>401</HttpCode> <Message>Unauthorized</Message> <InternalErrorCode xsi:nil="true"/> <ApiInfo> <Version>1.0</Version> <Timestamp>2019-04-04T09:01:11.407+01:00</Timestamp> <Provider>Liv-ex</Provider> </ApiInfo> </Response></pre>
--

5. API Listing

5.1 Invoices service (POST method)

Description

This service can be used to retrieve status and all related charges for the merchant’s sales invoices and sales credit notes. Furthermore, data related to pro forma invoices (for purchase transactions only) can be assessed by calling the service with ‘IxTradeNo’ request attribute.

Please note that this service does not provide any data for purchase invoices and purchase credit.

There are no mandatory fields. Calling the service without any attributes provided will result in all unpaid invoices for the last 2 years being returned. Two years limit applies to the ‘invoiceStatus’ request parameter. Calling the service with a single invoice number will allow to access the data for an invoice even if it is older than 2 years.

Base URI

[accounts/v1/invoices](#)

Request Parameters

Name	Mandatory	Description
invoiceStatus	N Default = 'not paid'	'paid', 'not paid', 'overdue' Type: alphanumeric <i>Only one value permitted per request</i> *not paid invoiceStatus includes overdue
paidTimeframe	N Default = '30' Applicable only to 'paid'	Determines the number of days to return data for 'paid' invoices <u>only</u> . Values: '30', '90', '180', '1year', '2year' Type: alphanumeric <i>Only one value permitted per request</i>
invoiceNo	N	Invoice number. Type: integer <i>Only one value permitted per request</i> 'paidTimeframe' and 'invoiceStatus' are being ignored when value provided
lxTradeNo	N	Type: alphanumeric. Example: '209978', 'lx209978', 'LX209978', 'Lx209978' <i>Only one value permitted per request</i> 'paidTimeframe' and 'invoiceStatus' are being ignored when value provided
merchantRef	N	The merchant's reference for the trade. Type: alphanumeric Max amount of characters is 30 <i>Only one value permitted per request</i> 'paidTimeframe' and 'invoiceStatus' are being ignored when value provided
lwin	N	LWIN 11/16/18. Type: alphanumeric <i>Only one value permitted per request</i> 'paidTimeframe' and 'invoiceStatus' are being ignored when value provided

Sample Request Body

JSON Request

```
{
  "invoices": {
    "invoiceStatus": "not paid",
    "paidTimeframe": "",
    "invoiceNo": "",
    "lxTradeNo": "",
    "merchantRef": "",
    "lwin": ""
  }
}
```

```
}
}
```

XML Request

```
<root>
<invoices>
<invoiceStatus>not paid</invoiceStatus>
<paidTimeframe></paidTimeframe>
<invoiceNo></invoiceNo>
<lXTradeNo></lXTradeNo>
<merchantRef></merchantRef>
<lwin></lwin>
</invoices>
</root>
```

Sample Response Body

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

Response parameters

Name	Description
merchantName	Type: alphanumeric
merchantAddress1	Type: alphanumeric
merchantAddress2	Type: alphanumeric
merchantPostcode	Type: alphanumeric
vatRate	Type: double
invoiceCombined	Type: alphanumeric Values: 'LX UK', 'LX EUR'
invoiceGoodsOnly	Type: alphanumeric Values: 'LX UK', 'LX EUR'
invoiceServicesOnly	Type: alphanumeric Values: 'LX UK', 'LX EUR'
invoiceNo	Type: integer Example: '209978'
invoiceDate	Type: alphanumeric, ISO 8601 Epoch time if JSON Example (JSON): 1549537950898 Example (XML): 2019-02-07T11:12:30
lXTradeNo	Type: integer Example: '209758', 'LX209758', 'lx209758', 'Lx209758'
merchantRef	Type: alphanumeric Example: '056400'
invoiceDescription	Type: alphanumeric Example: 209978 - Haut Brion 1995 12x75 1 unit(s) @ GBP 4,546 per unit L-WIN 101124719951200750
currency	Type: 3-character alphanumeric Example: 'GBP'
totalNetAmount	Type: double Example: '512.25'
totalIVAT	Type: double Example: '2.45'

totalInvoice	Type: double Example: '514.70'
paid	Type: Boolean true/false Example: false
isOverdue	Type: Boolean true/false Example: false
invoiceContent	Type: alphanumeric Values: 'servicesOnly', 'combined', 'goodsOnly'
chargeDescription	Type: alphanumeric Example: 209978 - Haut Brion 1995 12x75 1 unit(s) @ GBP 4,546 per unit L-WIN 101124719951200750
quantity	Type: integer
unitPrice	Type: double
netAmount	Type: double
VAT	Type: double

JSON Response

The response is sent per request.

```
{
  "status": "OK",
  "statusCode": "200",
  "message": "Request completed successfully.",
  "internalErrorCode": "R001",
  "apiInfo": {
    "version": "1.0",
    "timestamp": 1554365989969,
    "provider": "Liv-ex"
  },
  "merchantDetails": {
    "merchantName": "Company Ltd",
    "merchantAddress1": "Address 1",
    "merchantAddress2": "Address 2",
    "merchantPostcode": "L00 E00",
    "vatRate": 20,
    "invoiceCombined": "LX UK",
    "invoiceGoodsOnly": "LX EUR",
    "invoiceServicesOnly": "LX UK"
  },
  "invoices": [
    {
      "invoiceNo": 248899,
      "invoiceDate": 1538607600000,
      "lxTradeNo": 200000,
      "merchantRef": "m",
      "invoiceDescription": "200000 - Mondavi & Chadwick, Sena 2009 6x75 1 unit(s) @ GBP 390 per unit L-WIN 108321720090600750",
      "currency": "GBP",
      "totalNetAmount": 401.3,
      "totalVAT": 2.26,
      "totalInvoice": 403.56,
      "paid": false,
      "isOverdue": true,
      "invoiceContent": "combined",
      "charges": [
        {
          "chargeDescription": "200000 - Mondavi & Chadwick, Sena 2009 6x75 1 unit(s) @ GBP 390 per unit L-WIN 108321720090600750",
```

```

        "quantity": 1,
        "unitPrice": 390,
        "netAmount": 390,
        "vat": 0
    },
    {
        "chargeDescription": "200000 - Mondavi & Chadwick, Sena 2009 6x75
Commission on purchase at 2.00%",
        "quantity": 1,
        "unitPrice": 7.8,
        "netAmount": 7.8,
        "vat": 1.56
    },
    {
        "chargeDescription": "200000 - Mondavi & Chadwick, Sena 2009 6x75
Settlement fee on purchase @ GBP 3.5 per unit",
        "quantity": 1,
        "unitPrice": 3.5,
        "netAmount": 3.5,
        "vat": 0.7
    }
]
},
{
    "invoiceNo": 248500,
    "invoiceDate": 1538089200000,
    "lxTradeNo": null,
    "merchantRef": null,
    "invoiceDescription": "Vine charges invoice",
    "currency": "GBP",
    "totalNetAmount": 39.25,
    "totalVAT": 7.85,
    "totalInvoice": 47.1,
    "paid": false,
    "isOverdue": false,
    "invoiceContent": "servicesOnly",
    "charges": [
        {
            "chargeDescription": "Delivery Charge - Release 80000",
            "quantity": 1,
            "unitPrice": 21.4,
            "netAmount": 21.4,
            "vat": 4.28
        },
        {
            "chargeDescription": "Delivery Charge - Release 80000",
            "quantity": 1,
            "unitPrice": 17.85,
            "netAmount": 17.85,
            "vat": 3.57
        }
    ]
}
],
"errors": null
}

```

Invalid JSON response

```

{
    "status": "Bad Request",
    "statusCode": "400",
    "message": "Request was unsuccessful.",
    "internalErrorCode": "R000",
    "apiInfo": {
        "version": "1.0",
        "timestamp": 1554372879620,
    }
}

```



```

    "provider": "Liv-ex"
  },
  "merchantDetails": null,
  "invoices": null,
  "errors": {
    "error": [
      {
        "code": "V106",
        "message": "Invalid / incorrect invoiceStatus: [pad]. Possible values
are 'paid', 'not paid' or 'overdue'."
      },
      {
        "code": "V107",
        "message": "Invalid / incorrect paidTimeframe: [2 yr]. Possible values
are '30', '90', '180', '1year' or '2year'."
      }
    ]
  }
}

```

XML Response

The response is sent per request.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<accountInvoiceResponse 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>1.0</Version>
    <Timestamp>2019-04-04T11:13:04.372+01:00</Timestamp>
    <Provider>Liv-ex</Provider>
  </ApiInfo>
  <merchantDetails>
    <merchantName>Company Ltd</merchantName>
    <merchantAddress1>Address 1</merchantAddress1>
    <merchantAddress2>Address 2</merchantAddress2>
    <merchantPostcode>L00 E00</merchantPostcode>
    <vatRate>20.0</vatRate>
    <invoiceCombined>LX UK</invoiceCombined>
    <invoiceGoodsOnly>LX EUR</invoiceGoodsOnly>
    <invoiceServicesOnly>LX UK</invoiceServicesOnly>
  </merchantDetails>
  <invoices>
    <invoiceNo>248899</invoiceNo>
    <invoiceDate>2018-10-04T00:00:00.000+01:00</invoiceDate>
    <lxTradeNo>200000</lxTradeNo>
    <merchantRef>mditcham</merchantRef>
    <invoiceDescription>200000 - Mondavi & Chadwick, Sena 2009 6x75 1 unit(s) @
GBP 390 per unit L-WIN 108321720090600750</invoiceDescription>
    <currency>GBP</currency>
    <totalNetAmount>401.3</totalNetAmount>
    <totalVAT>2.26</totalVAT>
    <totalInvoice>403.56</totalInvoice>
    <paid>>false</paid>
    <isOverdue>>true</isOverdue>
    <invoiceContent>combined</invoiceContent>
    <charges>
      <charge>
        <chargeDescription>200000 - Mondavi & Chadwick, Sena 2009 6x75 1
unit(s) @ GBP 390 per unit L-WIN 108321720090600750</chargeDescription>
        <quantity>1</quantity>
      </charge>
    </charges>
  </invoices>
</accountInvoiceResponse>

```

```

        <unitPrice>390.0</unitPrice>
        <netAmount>390.0</netAmount>
        <vat>0.0</vat>
    </charge>
    <charge>
        <chargeDescription>200000- Mondavi & Chadwick, Sena 2009 6x75
Commission on purchase at 2.00%</chargeDescription>
        <quantity>1</quantity>
        <unitPrice>7.8</unitPrice>
        <netAmount>7.8</netAmount>
        <vat>1.56</vat>
    </charge>
    <charge>
        <chargeDescription>200000 - Mondavi & Chadwick, Sena 2009 6x75
Settlement fee on purchase @ GBP 3.5 per unit</chargeDescription>
        <quantity>1</quantity>
        <unitPrice>3.5</unitPrice>
        <netAmount>3.5</netAmount>
        <vat>0.7</vat>
    </charge>
</charges>
</invoices>
<invoices>
    <invoiceNo>248500</invoiceNo>
    <invoiceDate>2018-09-28T00:00:00.000+01:00</invoiceDate>
    <l1TradeNo xsi:nil="true"/>
    <merchantRef xsi:nil="true"/>
    <invoiceDescription>Vine charges invoice</invoiceDescription>
    <currency>GBP</currency>
    <totalNetAmount>39.25</totalNetAmount>
    <totalVAT>7.85</totalVAT>
    <totalInvoice>47.1</totalInvoice>
    <paid>false</paid>
    <isOverdue>false</isOverdue>
    <invoiceContent>servicesOnly</invoiceContent>
    <charges>
        <charge>
            <chargeDescription>Delivery Charge - Release 80000</chargeDescription>
            <quantity>1</quantity>
            <unitPrice>21.4</unitPrice>
            <netAmount>21.4</netAmount>
            <vat>4.28</vat>
        </charge>
        <charge>
            <chargeDescription>Delivery Charge - Release 80000</chargeDescription>
            <quantity>1</quantity>
            <unitPrice>17.85</unitPrice>
            <netAmount>17.85</netAmount>
            <vat>3.57</vat>
        </charge>
    </charges>
</invoices>
<invoices>
    <errors xsi:nil="true"/>
</accountInvoiceResponse>

```

Invalid XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<accountInvoiceResponse 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>1.0</Version>
    <Timestamp>2019-04-04T11:13:58.998+01:00</Timestamp>
    <Provider>Liv-ex</Provider>
  </ApiInfo>
  <merchantDetails xsi:nil="true"/>
  <errors>
    <error>
      <code>V106</code>
      <message>Invalid / incorrect invoiceStatus: [pad]. Possible values are
'paid', 'not paid' or 'overdue'.</message>
    </error>
    <error>
      <code>V107</code>
      <message>Invalid / incorrect paidTimeframe: [2 yr]. Possible values are
'30', '90', '180', '1year' or '2year'.</message>
    </error>
  </errors>
</accountInvoiceResponse>
```

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
V035	No records found
V058	Invalid / incorrect LWIN: [{v}]. Please provide a valid LWIN11, LWIN16 or LWIN18 code.
V100	Liv-Ex trade number: [value] is not available or does not exist.
V102	Merchant reference: [%s] is not available or does not exist.
V106	Invalid / incorrect invoiceStatus: [%s]. Possible values are 'paid', 'not paid' or 'overdue'.
V107	Invalid / incorrect paidTimeframe: [%s]. Possible values are '30', '90', '180', '1year' or '2year'.

V108	Liv-Ex invoice number: [%s] is not available or does not exist.
V109	Liv-Ex invoice number: [%s] does not correspond to the combination of the merchant reference / Liv-Ex trade number / LWIN code provided.
V110	Merchant reference: [%s] does not correspond to the combination of the Liv-Ex trade number / LWIN code provided.
V111	Liv-Ex trade number: [%s] does not correspond to the LWIN code.
V112	This service is available only for sales invoice and sales credit.
V113	For the request parameter [%s] only purchase transactions are available.

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.