LIV|EX

THE FINE WINE MARKET

Account Status API v2

Document Revision 1.0
Date of Issue: 20 February 2019
Date of revision: 20 February 2019

Daria Ershova

Table of Contents

1. Purpose

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

2. Glossary of Terms

Term	Meaning
Margin Deposit	Cash deposit held in the client trust account to support increased trading limit.
Available Buying Limit	Trading limit controls how much the client can buy on the Exchange before their trading is suspended.
Available Buying Headroom	Headroom on the Buying limit is a difference between the buying limit value and current exposure.
Available Selling Limit	Trading limit controls how much the client can sell on the Exchange before their trading is suspended.
Available Selling Headroom	Headroom on the Selling limit is the selling limit les any undelivered sales.
Available EP Selling Limit	Trading limit controls how much the client can sell on the Exchange using Standard En Primeur (SEP) contract before their trading is suspended.
Available EP Selling Headroom	Headroom on the EP Selling limit is a difference between the EP Selling limit value and total undelivered EP Sales.
EM Name	Name of the responsible Exchange Manager (EM).
VM Name	Name of the responsible Vine Manager (VM) from the logistics department.

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:
    - 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

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 merchant's 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. 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.

Example header (JSON)

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

Invalid header (JSON response)

```
{
    "status": "Unauthorized",
    "httpCode": "401",
    "message": "Unauthorized",
    "internalErrorCode": null,
    "apiInfo": {
        "version": "2.0",
        "timestamp": 1550225463393,
        "provider": "Liv-ex"
    }
}
```

Invalid header (XML response)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<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>2.0</Version>
        <Timestamp>2019-02-15T10:14:16.347Z</Timestamp>
        <Provider>Liv-ex</Provider>
    </ApiInfo>
</Response>
```

5. API Listing

5.1 Account Status service (GET method)

Description

This service can be used to retrieve a merchant's Liv-ex account status, trading privileges, balances, limits, membership package and exchange manager's contact details in either JSON or XML response formats.

Base URI

accounts/v2/accountStatus

Sample Response Body

The Account Status service will respond 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
currency	GBP, EUR Type: alphanumeric
currentBalance	Type: double
netAmountDue	Type: double
dueDate	Type: alphanumeric, ISO 8601
overdueAmount	Type: double
membershipPackageName	Bronze, Silver, Gold, Platinum, Black Type: alphanumeric
membershipChargeAmount	Monthly, Annually Type: double
membershipChargePeriod	Type: alphanumeric
membershipRenewalDate	Type: integer (JSON), alphanumeric (XML)
accountStatus	Live, Blocked, Closed Type: alphanumeric

tradingPrivilege	Full trading, No trading, Sell only Type: alphanumeric
releaseAllowed	Type: Boolean true/false
marginDeposit	Type: integer
availableByingHeadroom	Type: integer
availableBuyingLimit	Type: integer
availableSelling Headroom	Type: integer
availableSellingLimit	Type: integer
vintage	Type: integer
availableEPSelling Headroom	Type: integer
availableEPSellingLimit	Type: integer
emName	Type: alphanumeric
emTelephone	Type: alphanumeric
emEmail	Type: alphanumeric
vmName	Type: alphanumeric
vmTelephone	Type: alphanumeric
vmEmail	Type: alphanumeric

JSON Response

The response is sent per request.


```
{
    "status": "OK",
    "httpCode": "200",
    "message": "Request completed successfully.",
    "internalErrorCode": "R001",
    "apiInfo": {
        "version": "2.0",
        "timestamp": 1550228780829,
        "provider": "Liv-ex"
    },
    "accountStatus": {
        "currency": "GBP",
        "currentBalance": -4040,
        "netAmountDue": 0,
        "dueDate": null,
        "overdueAmount": 0,
        "membershipPackageName": "Silver",
        "membershipChargeAmount": 300,
        "membershipChargePeriod": "Monthly",
        "membershipRenewalDate": 1552176000000,
        "accountStatus": "Live",
        "tradingPrivilege": "No Trading",
        "releaseAllowed": true,
        "marginDeposit": 10000,
        "availableBuyingHeadroom": 117278,
        "availableBuyingLimit": 125000,
        "availableSellingHeadroom": 99630,
        "availableSellingLimit": 110000,
        "availableEPs": null,
        "emName": "Andrew Smith",
        "emTelephone": "",
        "emEmail": "andrew@liv-ex.com",
```

```
        "vmName": "Andrew Smith",
        "vmTelephone": "",
        "vmEmail": "andrew@liv-ex.com",
        "errors": null
    }
}
```


JSON response with invalid authentification


```
{
    "status": "Unauthorized",
    "httpCode": "401",
    "message": "Unauthorized",
    "internalErrorCode": null,
    "apiInfo": {
        "version": "2.0",
        "timestamp": 1550228343167,
        "provider": "Liv-ex"
    }
}
```


XML Response

The response is sent per request.


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<accountStatusResponse 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>2.0</Version>
        <Timestamp>2019-02-15T11:58:56.398Z</Timestamp>
        <Provider>Liv-ex</Provider>
    </ApiInfo>
    <accountStatus>
        <currency>GBP</currency>
        <currentBalance>-4040.0</currentBalance>
        <netAmountDue>0.0</netAmountDue>
        <dueDate xsi:nil="true"/>
        <overdueAmount>0.0</overdueAmount>
        <membershipPackageName>Silver</membershipPackageName>
        <membershipChargeAmount>300.0</membershipChargeAmount>
        <membershipChargePeriod>Monthly</membershipChargePeriod>
        <membershipRenewalDate>2019-03-10T00:00:00.000Z</membershipRenewalDate>
        <accountStatus>Live</accountStatus>
        <tradingPrivilege>No Trading</tradingPrivilege>
        <releaseAllowed>true</releaseAllowed>
        <marginDeposit>10000.0</marginDeposit>
        <availableBuyingHeadroom>117278.0</availableBuyingHeadroom>
        <availableBuyingLimit>125000.0</availableBuyingLimit>
        <availableSellingHeadroom>99630.0</availableSellingHeadroom>
        <availableSellingLimit>110000.0</availableSellingLimit>
        <availableEPs xsi:nil="true"/>
        <emName>Andrew Smith</emName>
        <emTelephone></emTelephone>
        <emEmail>andrew@liv-ex.com</emEmail>
        <vmName>Andrew Smith</vmName>
        <vmTelephone></vmTelephone>
        <vmEmail>andrew@liv-ex.com</vmEmail>
        <errors xsi:nil="true"/>
    </accountStatus>
```

```
</accountStatusResponse>
```

XML response with invalid authentification

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<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>2.0</Version>
        <Timestamp>2019-02-15T11:00:26.701Z</Timestamp>
        <Provider>Liv-ex</Provider>
    </ApiInfo>
</Response>
```

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.