

Overview

Purpose

To provide an overview of and user guide for the emsTradePoint API services.

Document Control

Record of version history.

Version	Date	Description
0.1	21/05/2018	Initial Draft
1.0	07/06/2018	Published document
1.1	26/06/2018	Included missing information for invoking OAuth2.0 tokens
1.2	10/08/2018	Included sample code for generating OAuth2.0 token
1.3	03/12/2018	Updated explanation for 429 error
1.4	09/02/2020	Updated oauth description, endpoint table, removed ruby samples
1.5	05/07/2021	Updated 'Service URI' typo and expanded 'Service Filters' section to include carbon filters.
1.6	23/05/2022	Updated endpoint list with /off_market_trades.

Document Approval

Approval for the current version.

Version	Date	Approved by
1.6		

If you have any queries regarding the content of this document, please contact the emsTradePoint Service Desk on either supportdesk@emstradepoint.co.nz or (04) 590-6692.

Request URIs and HTTP Headers

Service URIs

The base data URI is made up of the domain + API version, as follows:

- Production Domain | <https://exchange.emsTradePoint.co.nz>
- UAT Domain | <https://uat-exchange.emsTradePoint.co.nz>
- API Version | /api/v1

The base will be followed by an endpoint indicating a specific request or data set, i.e. /ticker, /orders, /trades. Each endpoint will have optional filters that may be used to refine a response.

Authentication

The emsTradePoint Exchange utilises OAuth 2.0 authentication, where credentials are supplied as a JSON payload and an access token is provided that must be provided with all data requests.

B2B Authentication

The requesting/Participant web server will need to call the following to request an access token:

- [{etpDomain}/oauth/token](#)

The following image demonstrates the JSON payload for both request and response. Note: the client_id and client_secret will be provided at the time of subscribing to the API suite.

```

1  POST [hostname]/oauth/token
2  // request payload for etp authentication
3  {
4  {
5      'payload': {
6          'grant_type': 'password',
7          'username': strUsername
8          'password': strPassword,
9          'client_id': strClientId,
10         'client_secret': strClientSecret
11     }
12 }
13 // response payload for authenticated user
14 {
15     'token_type': 'bearer',
16     'access_token': strAccessToken,
17     'refresh_token': strRefreshToken,
18     'created_at': strDateTime,
19     'expires_in': 7200
20 }

```

Third Party Authentication

Where a third-party client is requesting an access token, that client should call the following URI:

- [{etpDomain}/oauth/authorize](#)

Calling this URI will open an emsTradePoint interface which prompts the user to enter their username and password. On submission, the credentials are validated by the emsTradePoint authentication server. If successful, the authentication server will provide an access token that is valid for 2 hours.

In the instance where a third-party client is requesting an access token, a callback URI must be registered with the exchange which will redirect the client to a specific URL/web page after authentication.

- [{yourDomain}/oauth/callback](#)

Dedicated API Accounts

On request, the emsTradePoint Team can provide a 'read-only' account where the password does not expire.

Endpoint List

Company Info	<p><i>GET /v1/companies/ours</i></p> <p>Returns your company_id and user_id of your colleagues which can be used as optional filters for other endpoints.</p> <p>Additionally, when called by a broker, the response will include a list of customer for whom they can trade on behalf of.</p>
Products	<p><i>GET /v1/products</i></p> <p>Returns a list of active products available to trade on. Filters are available to refine the list.</p> <p><i>GET /v1/products/[id]</i></p> <p>Returns detail of the identified product, including bids/offers that make up the price stack.</p>
Orders	<p><i>GET /v1/orders</i></p> <p>Returns a list of gas orders that have been placed on the Exchange. Filters are available to refine the list.</p> <p><i>GET /v1/orders/nzu</i></p> <p>Returns a list of carbon orders that have been placed on the Exchange. Filters are available to refine the list.</p> <p><i>PATCH /v1/withdraw/all</i></p> <p>Enables the user to withdraw all live orders (both bids and offers) for the requesting company.</p>
Bids	<p><i>POST /v1/bids</i></p> <p>Submits a new bid to the stack. Bids terms can be submitted either as a JSON payload or in the URL.</p> <p><i>GET /v1/bids/[id]</i></p> <p>Returns details for a known id.</p> <p><i>PATCH /v1/bids/[id]/withdraw</i></p> <p>Enables the requestor to withdraw a bid with a known id.</p>

Offers	<p><i>POST /v1/offers</i> Submits a new offer to the stack. Offer terms can be submitted either as a JSON payload or in the URL.</p> <p><i>GET /v1/offers/[id]</i> Returns details for a known id.</p> <p><i>PATCH /v1/offers/[id]/withdraw</i> Enables the requestor to withdraw an offer with a known id.</p>
Trades	<p><i>GET /v1/trades</i> Returns a list of gas trades formed on the Exchange. Filters available to refine the list.</p> <p><i>GET /v1/trades/nzu</i> Returns a list of gas trades formed on the Exchange. Filters available to refine the list.</p>
Ticker	<p><i>GET /v1/ticker</i> Returns the 20 latest transactions (trades / orders / both) on the Exchange.</p> <p><i>GET /v1/ticker/trades</i> Returns the 20 latest trades on the Exchange.</p> <p><i>GET /v1/ticker/orders</i> Returns the 20 latest orders (either bids or offers) on the Exchange.</p>
Markets	<p><i>GET /v1/markets</i> Returns the current state of the product markets; On The Day (OTD), Day Ahead (DA), Carbon (NZU).</p>

Indices	<p><i>GET /v1/index</i></p> <p>Returns the latest monthly and quarterly indices (FRMI/FRQI) for the gas products. Filters are available to request a specific date range.</p> <p><i>GET /v1/index/nzu</i></p> <p>Returns the latest monthly and quarterly indices (ECMI/ECQI) for the carbon product. Filters are available to request a specific date range.</p>
Off Market Trades	<p><i>POST /v1/off_market_trades</i></p> <p>Enables submission of an Off-Market Trade.</p>

Service Filters

Pagination

By default, each GET request will return the latest data for the endpoint with default pagination of 30 records. This can be reduced by using the `per_page` filter but cannot be increased. When a response payload has been paginated, the response will include a “Link” header containing a payload similar to the below:

Link: <link-to-view-last-page>, rel=Last, <link-to-view-next-page>, rel=Next

Market Category

Most of the GET endpoints were designed to return gas market data by default. To return data relating to the carbon market, each request must include the `market_category=NZU` filter. To return both carbon and gas in the same response, the request must include the `market_category=NZU,NGP` filter.

Note: if the requesting user account does not have rights to view one or more of the requested markets, a 401 response will be returned.

Additional Filters

Many of the endpoints provided will allow additional filtering to replicate existing functionality available through the web application. These filters are outlined in the live SwaggerAPI documentation.

API Documentation

Live API documentation and a test interface can be found on the following URL:

- <https://uat-exchange.emstradepoint.co.nz/api/docs>

The documentation provides current API schemas for the requests and responses. If you wish to run beta testing on the API suite with live data, please contact emsTradePoint to arrange a temporary account for the user acceptance platform.

Response Headers and Data

The following status codes are supported by the emsTradePoint API suite.

HTTP Status	HTTP Headers	HTTP Body and explanation
200	Access-Control-Allow-Origin: Access-Control-Request-Method: Cache-Control: Connection: Content-Length: Date: [date and time] ETag: Server: Status: Strict-Transport-Security: Vary: X-Request-Id: X-Runtime: X-Version:	<pre>{ [data with one or more property] }</pre> <p><i>A successful GET request for data or PATCH request to update data. One or more data properties will be present containing data.</i></p>
201	Access-Control-Allow-Origin: Access-Control-Request-Method: Cache-Control: Connection: Content-Length: Date: [date and time] ETag: Server: Status: Strict-Transport-Security: Vary: X-Request-Id: X-Runtime: X-Version:	<pre>{ [data with one or more property] }</pre> <p><i>A successful POST request / submission of data. One or more data properties will be present containing data.</i></p>

HTTP Status	HTTP Headers	HTTP Body and explanation
400	Access-Control-Allow-Origin: Access-Control-Request-Method: Cache-Control: Connection: Content-Length: Date: [date and time] Server: Status: Strict-Transport-Security: Vary: X-Request-Id: X-Runtime: X-Version:	<pre data-bbox="619 483 1136 568">{ "error": [message identifying problem input value] }</pre> <p data-bbox="619 609 1350 667"><i>The service client's request has invalid inputs, e.g. incorrect date format, text in place of integer.</i></p>
401	Access-Control-Allow-Origin: Access-Control-Request-Method: Cache-Control: Connection: Content-Length: Date: [date and time] Server: Status: Strict-Transport-Security: Vary: X-Request-Id: X-Runtime: X-Version:	<pre data-bbox="619 994 1046 1079">{ "error": [authentication specific message] }</pre> <p data-bbox="619 1120 1382 1178"><i>The service client failed to include a current or valid authentication token in the request</i></p>

HTTP Status	HTTP Headers	HTTP Body and explanation
422	Access-Control-Allow-Origin: Access-Control-Request-Method: Cache-Control: Connection: Content-Length: Date: [date and time] Server: Status: Strict-Transport-Security: Vary: X-Request-Id: X-Runtime: X-Version:	<pre data-bbox="619 483 1157 571">{ "Error": "Validation failed: [detail of failed criterion]" }</pre> <p data-bbox="619 611 1377 672"><i>The service failed due to a validation error; exceeded exposure/trader limit, incorrect user/organisation role, etc).</i></p>
429	Access-Control-Allow-Origin: Access-Control-Request-Method: Cache-Control: Connection: Content-Length: Date: [date and time] Server: Status: Strict-Transport-Security: Vary: X-Request-Id: X-Runtime: X-Version:	<pre data-bbox="619 994 933 1081">{ "Error": "Rate Limit Exceeded" }</pre> <p data-bbox="619 1122 1369 1182"><i>The service failed due to a validation error; exceeded current 'requests per minute' limit for the endpoint).</i></p>