

VEC Report B2B Instructions for Customers

Purpose:

This is a document to instruct how customer applications can access the VEC Report data directly via API URL calls, thus bypass accessing the VEC Report page on the UI front end.

Instructions

To pull from the GCTS API end point URLs, you will need to authenticate first.

1) Authenticate by making a HTTP POST request to the security API end point:

<https://secure2.gcts.tcenergy.com/security/authenticate>

- Request header needs to contain:
 - Content-Type: application/json
- Content Body needs to contain your username and password:
 - { "username": "<user>", "password": "<pwd>" }
 - **Note:** As this is passed over via HTTPS protocol, the user and pwd will be encrypted

The result should return a HTTP 200 OK, and the content look as follows:

```
{
  "tokenType": "Bearer",
  "accessToken": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsInR5cCI6IWRX",
  "refreshToken": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsInR5cCI6IWRX"
}
```

This shows that you have authenticated successfully, and the results return a JWT **access token** and refresh token. You will need the value of the **access token** to make calls to the reporting end points. The token value is an encoded string that contains information on what sort of permissions you have. This is needed so that the end point can authorize you first before serving your request.

2) Once you have the JWT token, make a HTTP GET request to the reporting end point:

<https://secure2.gcts.tcenergy.com/flowinggas/pipelines/ML/reports/volumeEnergyComposition/export?fromGasDay=<From Gas Day>&toGasDay=<To Gas Day>&reportByType=Daily&sinceDate=<Since Date>&sinceTime=<Since Time>&stationNumbers=<Station Numbers>>

- Request header needs to contain:
 - Authorization: Bearer <JWT access token value>
 - Content-Type: application/json
- **From Gas Day** and **To Gas Day** make up the date range of the report and will need to be in the following format:
 - YYYY-MM-DD

- Ex) 2020-01-01
- To get the hourly report, replace **reportByType=Daily** with **reportByType=Hourly**
- **Since Date** is used to return the measurement adjustments made since the provided date. It will need to be in the following format:
 - YYYY-MMM-DD
 - Ex) 2020-Jan-01
 - **Note:** this is an optional field, and to be used in conjunction with Since Time
- **Since Time** is used to return the measurement adjustments made since the provided time. It will need to be in the following format:
 - HH:MM (24-hour clock)
 - Ex) 23:00
 - Default should be 00:00
 - **Note:** this is an optional field, and to be used in conjunction with Since Date
- **Station Numbers** is a list of one or more measurement station numbers of which you are an Operator or Agent of
 - Ex) 50011,50012,50053
 - **Note:** this is a required field

Submitting this call should return a HTTP 200 OK, and the content looks as follows:

```

)}}',
[{"stationNumber":51999,"stationName":"Empress","stationType":{"id":"Rec Meas","description":"Receipt Measurement"},"gasDay":"2020-01-12T00:00:00","fromDateTime":"2020-01-12T09:00:00","toDateDateTime":"2020-01-13T08:00:00","energy":2701711.1338041,"volume":70131.7112028,"volumeUnitOfMeasure":null,"pressure":null,"pressureUnitOfMeasure":"kPa","temperature":null,"temperatureUnitOfMeasure":"degC","acceptabilityIndicator":50,"sg":null,"heatValue":38.5216165,"heatValueUnitOfMeasure":"MJ/m3","n2":null,"c02":null,"c1":null,"c2":null,"c3":null,"nC4":null,"iC4":null,"nC5":null,"iC5":null,"nC6":null,"nC7":null,"nC8":null,"he":null}}]

```

The response returned is in JSON format. As JSON could be susceptible to Cross Site Request Forgery, we have protected the response by prefixing the following characters to it: **}}}**,

For more information on JSON Vulnerability, refer to the following:

[https://docs.angularjs.org/api/ng/service/\\$http#json-vulnerability-protection](https://docs.angularjs.org/api/ng/service/$http#json-vulnerability-protection)

<https://haacked.com/archive/2008/11/20/anatomy-of-a-subtle-json-vulnerability.aspx/>

3) How to work with the JSON response:

You will need to parse the JSON response. Due to the JSON security prefix, you will first need to strip it

out before you can parse the JSON data. Note that some frameworks will automatically strip it for you (for example, AngularJS and Angular). Otherwise, you will need to manually remove it from the string.

Environment URLs:

Qual 2 Test Environment URLs:

<https://secure2.gcts-qual2.tcenergy.com/security/authenticate>

<https://secure2.gcts-qual2.tcenergy.com/flowinggas/pipelines/ML/reports/volumeEnergyComposition/export>

Production Environment URLs:

<https://secure2.gcts.tcenergy.com/security/authenticate>

<https://secure2.gcts.tcenergy.com/flowinggas/pipelines/ML/reports/volumeEnergyComposition/export>