1 Authentication

To get access to the API you will need a token.

This is created by providing the authentication endpoint with your username & password as given to you.

Below (example 1) is javascript code using ajax to get the token:

Example 1: Javascript to access the token for the platform API

```
function authenticate() {
   var token;
   $.ajax({
      url: 'https://api.argandsolutions.com/v1/auth/login',
      method: 'POST',
      async: false,
      data: {
         "email": "<your email address – case sensitive",
         "password": "<your password>"
      },
      complete: function(response) {
         $('#output').html(response.responseText);
         var loginObj = $.parseJSON(response.responseText);
         token = loginObj.token;
      },
   });
   return token;
}
```

2 Parsing the Token into API Calls

Once you have the token, this can be parsed into the API calls using the following method (example 2)

Example 2: Header used for token authorisation

```
headers: {'X-Auth-Token': token}
```

3 Getting Available Locations

Available locations for your authorised access can be achieved using the following API call

Method: GET
URL: https://api.argandsolutions.com/v1/locations

The JSON returned contains all the location IDs and names that you have access to.
The ID is required for further API calls.

You can parse the data for ONLY the parent location ID by filtering for

parent_id == null

4 Getting Meter Data For a Location ID

Once you have the location IDs you can then use this to get all of the meter IDs and the information about each meter by using the following GET request:

Method = GET
https://api.argandsolutions.com/v1/locations/<location_id>/generation-meters

Please ignore the fact that it has “generation-meters” at the end => this will bring back all meters within a location Id with an example structure shown below:

```
[
 {
  "id": <meter_id>,
  "name": <meter name >,
  "client": {
   "id": <Client ID>,
   "name": < Client Name>
  },
  "location": {
   "id": <Location ID>,
   "name": <Location Name>
  },
  "meter_type": {
   "id": <Meter Type ID>,
   "name": <Meter Type>
  },
  "energy_type": {
   "id": <Meter Energy Type ID>,
   "name": <Energy Type>
  },
  "site_details": [],
  "energy_supplier": {
   "id": "1",
```

```
    "name": "Good Energy B2B"
   },
   "data_interval": 0.5,
   "meter_model": <entered model type>
   "meter_manufacturer": <entered meter manufacturer>
   "is_export_deemed": "no",
   "fit_submission": "no",
   "energy_supplier_id": "1",
   "offset": "0",
   "serial_id": null,
   "deemed_export_rate": null,
   "fit_client_id": null,
   "fit_accreditation_id": null,
   "fit_client_rgs_id": null,
   "fit_expiry_date": null,
   "fit_first_read_date": null,
   "fit_generator_id": null,
   "fit_meter_read_frequency": null
  },
Etc etc.
]
```


5 Energy Meter Reading API

5.1 Single Meter Reading

For a given location ID (locID) you can get the cumulative meter reading for a specified timestamp using the following API

Method: GET
URL: https://api.argandsolutions.com/v1/locations/<locID>/meter-readings?dateto=<endUTCTimestampInMilliSeconds>

Where,

<locID> = locationID from location API call
<endUTCTimestampInMilliSeconds> = ending UTC timestamp in milliseconds

5.2 Cumulative Energy Between Timestamps

To get the cumulative energy between specified timestamps and to get the start & end readings for this period, please use the following API call

Method: GET
URL: https://api.argandsolutions.com/v1/locations/<locID>/meter-readings?datefrom=<startUTCTimestampInMilliSeconds&dateto=<endUTCTimestampInMilliSeconds>

Where,

<locID> = locationID from location API call
<startUTCTimestampInMilliSeconds> = starting UTC timestamp in milliseconds
<endUTCTimestampInMilliSeconds> = ending UTC timestamp in milliseconds

5.3 Cumulative Energy Readings Between Timestamps for a Specific Meter ID

To get the cumulative energy readings for a specific meter then you can use the following API call:

Method: GET
URL: https://api.argandsolutions.com/v1/meters/<meter_id>/cumulative-energy-readings?timestamp_from=<start_UTC_timestamp>&timestamp_to=<end_UTC_timestamp>

Where,

<meter_id> = meter ID
<start_UTC_timestamp> = timestamp in UTC milliseconds
<end_UTC_timestamp> = timestamp in UTC milliseconds

This returns the cumulative energy readings between the 2 entered UTC dates.

5.4 Net Energy & Average (30 minute) Power Values

To get individual timestamped net energy use and power (average for 30 minute period) you can use the following API call

Method: GET
URL:
https://api.argandsolutions.com/v1/locations/<locID>/realtime?type=energy&lens=export-onsite-use&direction=generation&datefrom=<startUTCTimestampInMilliSeconds>&dateto=<endUTCTimestampInMilliSeconds >&chart=heatmap

Where,

<locID> = locationID from location API call
<startUTCTimestampInMilliSeconds> = starting UTC timestamp in milliseconds
<endUTCTimestampInMilliSeconds> = ending UTC timestamp in milliseconds

6 Reactive Power Values for a Specified Time Period

To get individual timestamped reactive power (taken at each 30 minute period) you will need to go through a number of specific steps which are documented as follows:

6.1 Get available power quality meters & register IDs for location

You need to start by getting the following:

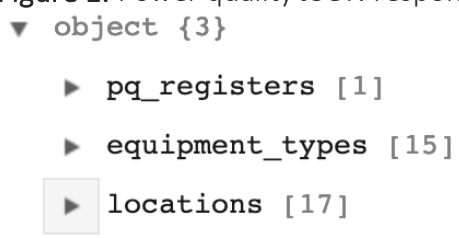
Method: GET
URL: https://api.argandsolutions.com/v1/locations/'+locationId+'/power-quality/headings?datefrom='+start+'&dateto='+end

Where,

<locID> = locationID from location API call
<startUTCTimestampInMilliSeconds> = starting UTC timestamp in milliseconds
<endUTCTimestampInMilliSeconds> = ending UTC timestamp in milliseconds

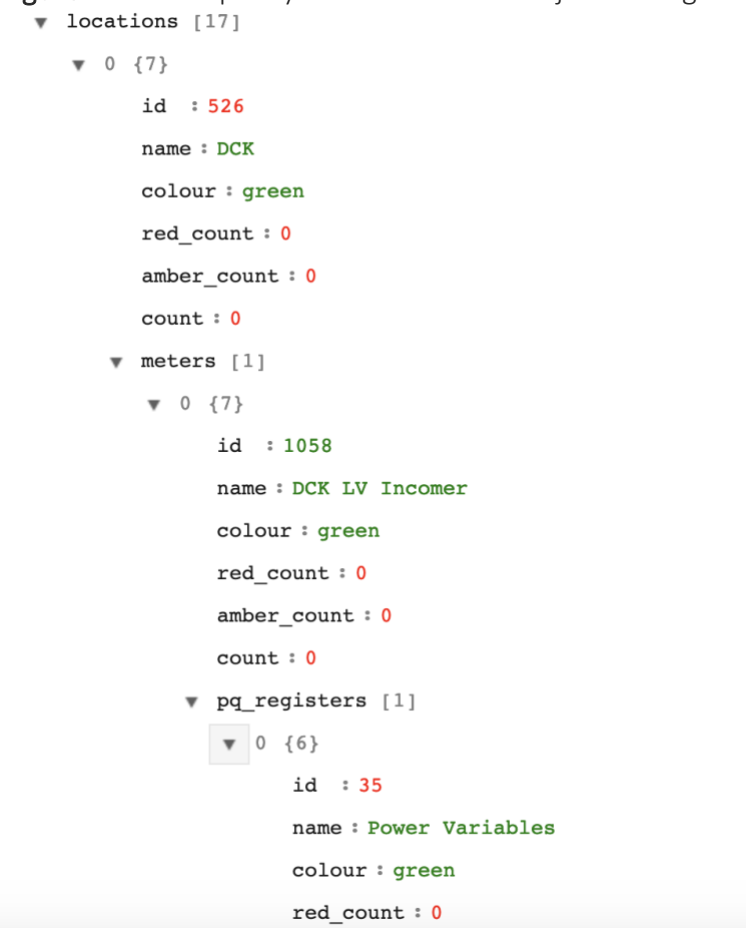
This will return a JSON object structured as shown in figure 1 below.

Figure 1: Power quality JSON response



Access the locations key to then find the available meters, their IDs and then the available power quality register IDs i.e. Reactive Power (kVA) using the object structure shown in figure 2.

Figure 2: Power quality > locations JSON object through to the meters and regsiters



From the object you will need to collect the meter and register IDs required.

6.2 Get Timestamped Power Quality Data

With the meter IDs and register IDs you will then be able to get interval data for the required period (between a start and end timestamp)

This can be done with the following API

Method: GET
URL: https://api.argandsolutions.com/v1/meters/<meterID>/power-quality/<registerID>/readings?datefrom=<UTCStart>&dateto=<UTCEnd>

Where,

<meterID> = meter ID from power quality API call
<registerID> = regsiterID for specific data point from power quality API call
<UTCStart> = starting UTC timestamp in milliseconds
<UTCEnd> = ending UTC timestamp in milliseconds

7 Data Services

You access the data in your data services using the following GET requests:

1. Get all results for a specific data service id (enter ID number in place of <id>)
    a. API Type = GET.
    b. Set headers: {'X-Auth-Token': token}
    c. https://api.argandsolutions.com/v1/client-data-services/1/schedule/id/results?limit=5
    d. This will give you a list of the latest sub-results for the data service
    e. You can sort by the timestamp when they were created to get the latest
    f. Get the latestResultId that you wish to access
2. Get the results for the chosen latestResultId
    a. API Type = GET.
    b. Set headers: {'X-Auth-Token': token}
    c. https://api.argandsolutions.com/v1/client-data-services/1/schedule/id/results/latestResultId
    d. You can then look through the array to get data in types =
        i. Files
        ii. Charts
        iii. Tables
        iv. HTML reports
    e. Using the keys in the array / objects