Accessing the PowerFlex API

Our APIs use json web tokens (JWT) for authentication. To obtain your JWT, send an https post request. In all examples below, the url will be based on a host called `host`, which results in `host.powerflex.com`. This will be specific to your organization. Also, port 9443 will be used, unless stated otherwise.

```
curl -X POST https://host.powerflex.com:9443/login -H 'cache-control: no-cache' -H
'content-type: application/json' -d '{"username":"<your username>", "password":"<your
password>"}'
```

If your login credentials are accepted, the `access_token` will be returned in json format. It will be valid for 1 day. After 1 day, you will need to login again to obtain a new token.

```
{"access_token":"eyJhbGciOiJIUzI1NiIsIn.eyJqdGkiOii0yN2UxLTQwMNzg0MjY3YzkwMjciLCJleHAiOjE
1NjYyZXNoIjpmYWxzZSwiaWF0IjoxNTY1OTgyMTExLCJ0eXBlIjoiYWNjZXNzIiwibmJmIjoxNTY1OTgyMTExLCJp
ZGVudGl0eSI6InBvd2VyZmxleCJ9.Zbcx3kJjOF3ZrTe3Jzw6eQ2VhXTDEW2GxM3ZR5pufCc"}
```

The `access_token` will then be included in the `Authorization` http header. The header should look like this:

```
Authorization: Bearer <access_token>
```

Note the space between `Bearer` and `access_token`.

API Endpoints

Endpoint url variables such as `acn_id` and `acc_id` are sometimes needed. These variables are used to specify which site to query. If you do not know these values, please contact PowerFlex.

Time-Series Data

Time-series data is made available to you via the `get_measurement_data` API endpoint. This data is stored in InfluxDB databases: `ct_response`, `evse_request`, and `evse_response`.

The `ct-response` measurement is described in the table below.

Field	Type	Description
time	*float*	Unix epoch time of a data point.
acn_id	*string*	PowerFlex internal ID that represents a customer.
acc_id	*string*	PowerFlex internal ID that represents a parking garage or building.
acg_id	*string*	PowerFlex internal ID that represents a group of chargers.
acs_id	*string*	PowerFlex internal ID taht represents a charging station.
charging_state	*string*	`"UNPLUGGED"`, `"ADAPTIVE"`, `"IDLE"`, `"READY"`, or `"NOT CHARGING"`.
energy_delivered	*float*	Energy delivered so far during the current session in mWh (milli-Watt-hour).
evse_address	*string*	PowerFlex internal ID made up of `acn_id`, `acc_id`, `acg_id`, and `acs_id`.
evse_type	*string*	The type of charging station. e.g., `"AeroVironment"` or `"Tesla"`.
mamps_actual	*float*	The actual charging current in mA.
peak_rate	*float*	An internal variable used to calculate `mamps_rampdown` in the `evse_request` measurement.
phases	*string*	`"AB"`, `"BC"` or `"CA"`. These represent which phases the EVSE is on in a 3-phase power system.
pilot_actual	*float*	The pilot signal current sent to the charging stations in A.
power	*float*	Power in kW.
space_number	*string*	Parking space number.
version	*string*	Version of this measurement.
voltage	*float*	The voltage across the two hots in mV

The `evse_request` measurement is described in the table below.

Field	Type	Description
time	*float*	Unix epoch time of a data point.
acn_id	*string*	PowerFlex internal ID that represents a customer.
acc_id	*string*	PowerFlex internal ID that represents a parking garage or building.
acg_id	*string*	PowerFlex internal ID that represents a group of chargers.
acs_id	*string*	PowerFlex internal ID taht represents a charging station.
charging_state	*string*	`"UNPLUGGED"`, `"ADAPTIVE"`, `"IDLE"`, `"READY"`, or `"NOT CHARGING"`.
evse_address	*string*	PowerFlex internal ID made up of `acn_id`, `acc_id`, `acg_id`, and `acs_id`.
evse_type	*string*	The type of charging station. e.g., `"AeroVironment"` or `"Tesla"`.
mamps_last	*float*	The last charging current in mA.

Field	Type	Description
mamps_limit	*float*	The pilot signal current sent to the charging stations in mA.
mamps_rampdown	*float*	An estimate of the car's maximum charging current in mA.
space_number	*string*	Parking space number
version	*string*	Version of this measurement

The `evse_response` measurement is described in the table below.

Field	Type	Description
time	*float*	Unix epoch time of a data point.
acn_id	*string*	PowerFlex internal ID that represents a customer.
acc_id	*string*	PowerFlex internal ID that represents a parking garage or building.
acg_id	*string*	PowerFlex internal ID that represents a group of chargers.
acs_id	*string*	PowerFlex internal ID taht represents a charging station.
charging_state	*string*	`"UNPLUGGED"`, `"ADAPTIVE"`, `"IDLE"`, `"READY"`, or `"NOT CHARGING"`.
connected	*float*	`1` if the car is connected or `0` otherwise.
contactor	*float*	`1` if the EVSE contactor is closed or `0` otherwise.
evse_address	*string*	PowerFlex internal ID made up of `acn_id`, `acc_id`, `acg_id`, and `acs_id`.
evse_type	*string*	The type of charging station. e.g., `"AeroVironment"` or `"Tesla"`.
mamps_last	*float*	The last charging current in mA.
response_period	*float*	Elapsed time since the last report to this measurement.
space_number	*string*	Parking space number
version	*string*	Version of this measurement

get_measurement_data

Url: `/get_measurement_data`
Method: POST

This endpoint will return data in json format. An example of the returned data format can be found here: https://docs.influxdata.com/influxdb/v1.7/tools/api/#examples-2

JSON data to send:

Key	Type	Description
measurement	*string*	One of the following: `ct_response`, `evse_request`, or `evse_response`

Key	Type	Description
time_filter	*array of 2 integers*	`[start_time, stop_time]`. These should be in seconds from epoch.

Example:
The following query will get all datapoints in `evse_response` where `1566333668 <= time < 1566344979`.

```
curl -X POST https://host.powerflex.com:9443/get_measurement_data -H 'Authorization:
Bearer
eyJhbGciOiJIUzI1NiIsIn.eyJqdGkiOii0yN2UxLTQwMNzg0MjY3YzkwMjciLCJleHAiOjE1NjYyZXNoIjpmYWxz
ZSwiaWF0IjoxNTY1OTgyMTExLCJ0eXBlIjoiYWNjZXNzIiwibmJmIjoxNTY1OTgyMTExLCJpZGVudGl0eSI6InBvd
2VyZmxleCJ9.Zbcx3kJjOF3ZrTe3Jzw6eQ2VhXTDEW2GxM3ZR5pufCc' -H 'cache-control: no-cache' -H
'content-type: application/json' -d '{"measurement":"evse_response",
"time_filter":[1566333668, 1566344979]}'
```

External Signals

External signals are how you can control how much power you want allocated to the charging stations at your site.

For example, if you set the external signal to 100 kW, then the aggregate power consumption of all PowerFlex controlled charging stations will be limited to 100 kW.

If you want to set a time-varying max power, please contact PowerFlex directly.

get_external_signal

Url: `/get_external_signal/<int:acc_id>`
Method: POST

This endpoint will return details of any external signals that belong to the site.

JSON data to send:

Key	Type	Description
name	*string, optional*	If provided, `get_external_signal` will return the data for the external signal named. Otherwise, all external signals will be returned.

Example 1.
`name` is unspecified:

```
curl -X POST https://host.powerflex.com:9443/get_external_signal/1 -H 'Authorization:
Bearer
eyJhbGciOiJIUzI1NiIsIn.eyJqdGkiOii0yN2UxLTQwMNzg0MjY3YzkwMjciLCJleHAiOjE1NjYyZXNoIjpmYWxz
ZSwiaWF0IjoxNTY1OTgyMTExLCJ0eXBlIjoiYWNjZXNzIiwibmJmIjoxNTY1OTgyMTExLCJpZGVudGl0eSI6InBvd
2VyZmxleCJ9.Zbcx3kJjOF3ZrTe3Jzw6eQ2VhXTDEW2GxM3ZR5pufCc' -H 'cache-control: no-cache' -H
'content-type: application/json' -d '{}'
```

returns

```
{"DCM_EVSE_Allocation":{"is_enabled":true,"max_load":21.616,"max_load_unit":"kW","name":"
DCM_EVSE_Allocation"}}
```

Example 2.
`name` is specified as `DCM_EVSE_Allocation`.

```
curl -X POST https://host.powerflex.com:9443/get_external_signal/1 -H 'Authorization:
Bearer
eyJhbGciOiJIUzI1NiIsIn.eyJqdGkiOii0yN2UxLTQwMNzg0MjY3YzkwMjciLCJleHAiOjE1NjYyZXNoIjpmYWxz
ZSwiaWF0IjoxNTY1OTgyMTExLCJ0eXBlIjoiYWNjZXNzIiwibmJmIjoxNTY1OTgyMTExLCJpZGVudGl0eSI6InBvd
2VyZmxleCJ9.Zbcx3kJjOF3ZrTe3Jzw6eQ2VhXTDEW2GxM3ZR5pufCc' -H 'cache-control: no-cache' -H
'content-type: application/json' -d '{"name":"DCM_EVSE_Allocation"}'
```

returns

```
{"is_enabled":true,"max_load":21.616,"max_load_unit":"kW","name":"DCM_EVSE_Allocation"}
```

set_external_signal

Url: `/set_external_signal/<int:acc_id>`
Method: POST

This endpoint allows you to set up a new external signal or change an existing one.

JSON data to send:

Key	Type	Description
max_load	*float*	The maximum aggregate power in kW or A
max_load_unit	*string*	`kW` or `A`
is_enabled	*boolean*	`true` to enable and `false` otherwise

Example:

```
curl -X POST https://host.powerflex.com:9443/set_external_signal/1 -H 'Authorization:
Bearer
eyJhbGciOiJIUzI1NiIsIn.eyJqdGkiOii0yN2UxLTQwMNzg0MjY3YzkwMjciLCJleHAiOjE1NjYyZXNoIjpmYWxz
ZSwiaWF0IjoxNTY1OTgyMTExLCJ0eXBlIjoiYWNjZXNzIiwibmJmIjoxNTY1OTgyMTExLCJpZGVudGl0eSI6InBvd
2VyZmxleCJ9.Zbcx3kJjOF3ZrTe3Jzw6eQ2VhXTDEW2GxM3ZR5pufCc' -H 'cache-control: no-cache' -H
'content-type: application/json' -d '{"max_load":21.6, "max_load_unit":"kW",
"is_enabled": true, "name":"DCM_EVSE_Allocation"}'
```

returns the same data back if successful

```
{"is_enabled":true,"max_load":21.6,"max_load_unit":"kW","name":"DCM_EVSE_Allocation"}
```

Demand Charge Mitigation

Demand Charge Mitigation (DCM) is a PowerFlex data analytics feature to prevent additional demand charges due to electric vehicle charging. If DCM is enabled at your site, you can use the following API endpoints to supply the additional information required for DCM to properly function.

set_real_time_load

Url: `/set_real_time_load/<int:acc_id>`
Method: POST

Set the real-time load of the site using this endpoint. It can be either site + EVSEs or just the site (this is defined during DCM setup).

JSON data to send:

Key	Type	Description
real_time_load_kW	*integer*	The real-time load in kW

Example:

```
curl -X POST https://host.powerflex.com:9443/set_real_time_load/2 -H 'Authorization:
Bearer
eyJhbGciOiJIUzI1NiIsIn.eyJqdGkiOii0yN2UxLTQwMNzg0MjY3YzkwMjciLCJleHAiOjE1NjYyZXNoIjpmYWxz
ZSwiaWF0IjoxNTY1OTgyMTExLCJ0eXBlIjoiYWNjZXNzIiwibmJmIjoxNTY1OTgyMTExLCJpZGVudGl0eSI6InBvd
2VyZmxleCJ9.Zbcx3kJjOF3ZrTe3Jzw6eQ2VhXTDEW2GxM3ZR5pufCc' -H 'cache-control: no-cache' -H
'content-type: application/json' -d '{"real_time_load_kW": 1100}'
```

get_target_peak_power_kW

Url: `/get_target_peak_power_kW/<int:acc_id>`
Method: GET

Get the target peak power in kW for each month. DCM will try to limit the site + EVSE aggregate power to these values.

Example:

```
curl -X GET https://host.powerflex.com:9443/get_target_peak_power_kW/2 -H 'Authorization:
Bearer
eyJhbGciOiJIUzI1NiIsIn.eyJqdGkiOii0yN2UxLTQwMNzg0MjY3YzkwMjciLCJleHAiOjE1NjYyZXNoIjpmYWxz
ZSwiaWF0IjoxNTY1OTgyMTExLCJ0eXBlIjoiYWNjZXNzIiwibmJmIjoxNTY1OTgyMTExLCJpZGVudGl0eSI6InBvd
2VyZmxleCJ9.Zbcx3kJjOF3ZrTe3Jzw6eQ2VhXTDEW2GxM3ZR5pufCc' -H 'cache-control: no-cache'
```

returns

```
{"Apr":200,"Aug":200,"Dec":200,"Feb":123,"Jan":200,"Jul":200,"Jun":200,"Mar":200,"May":20
0,"Nov":200,"Oct":200,"Sep":200}
```

set_target_peak_power_kW

Url: `/set_target_peak_power_kW/<int:acc_id>`
Method: POST

Set the target peak power in kW for each month. DCM will try to limit the site + EVSE aggregate power to these values.

JSON data to send:

Key	Type	Description
target_peak_power	*dict*	A dictionary of the monthly peak power in kW. The keys should be month abbreviations and the values should be integers. You can just the months you want to modify.

Example:

```
curl -X POST https://host.powerflex.com:9443/set_target_peak_power_kW/2 -H
'Authorization: Bearer
eyJhbGciOiJIUzI1NiIsIn.eyJqdGkiOii0yN2UxLTQwMNzg0MjY3YzkwMjciLCJleHAiOjE1NjYyZXNoIjpmYWxz
ZSwiaWF0IjoxNTY1OTgyMTExLCJ0eXBlIjoiYWNjZXNzIiwibmJmIjoxNTY1OTgyMTExLCJpZGVudGl0eSI6InBvd
2VyZmxleCJ9.Zbcx3kJjOF3ZrTe3Jzw6eQ2VhXTDEW2GxM3ZR5pufCc' -H 'cache-control: no-cache' -H
'content-type: application/json' -d '{"target_peak_power": {"Jun": 700, "Jul": 555}}'
```

returns

```
{"Apr":200,"Aug":200,"Dec":200,"Feb":123,"Jan":200,"Jul":555,"Jun":700,"Mar":200,"May":20
0,"Nov":200,"Oct":200,"Sep":200}
```