

ChargeLab API reference

Core API version 1.12 • Document date: Apr 29, 2026

The ChargeLab API is intended to allow the caller to query available electric vehicle Chargers connected to ChargeLab's Charging Station Management System (CSMS). You can list charger availability, start and stop charging sessions, and query the transactions generated by those charging sessions.

Terminology

The following terminology is used throughout this API Reference document:

EV: A battery electric vehicle (BEV) or plug-in hybrid electric vehicle (PHEV).

Charger: An electric vehicle charger, also called a "Charge Point" (OCPP 1.6, OCPI 2.2) or "Charging Station" (OCPP 2.x). Within the client-server architecture of OCPP, the EV charger is the client while ChargeLab's CSMS is the server. A Charger is typically a single device with a central microprocessor, modem/communication board, metal or plastic enclosure, and one or more charging ports. ChargeLab's system supports various [Type 1](#) and [Type 2](#) charging ports, both AC and DC. There is support for the [Tesla \(NACS\)](#) port as well.

CSMS: ChargeLab's Charging Station Management System. The cloud-based server built and maintained by ChargeLab to which OCPP EV chargers connect. Also called a Central System (OCPP 1.6).

Port: The part of a Charger responsible for connecting to and charging a single EV. A Charger will always have at least one Port but may have more. OCPP 1.6 refers to Ports as "Connectors", while OCPP 2.x and OCPI 2.2 refer to Ports as EVSEs (Electric Vehicle Supply Equipment). A Port typically has one ConnectorType to make a physical connection to the EV (e.g. J1772, J1772 CCS, CHAdeMO, or NACS). However, it is possible for a Port to have multiple types of connectors, but only one can be in use at a time.

Session: A charging session attempt. If successful, it will result in an associated Transaction being created.

Transaction: The record of a successful Session, including details on energy delivered, when the session started and ended, plus other information related to charging.

Table of contents

Terminology	1
API basics	2
URLs	3
Authentication	3
Payloads	3
HTTP return codes	3
DateTime formats	3
Role	4
Filtering	4
Pagination	5
Charger entity	7
Role	7
Public access to charger information	7
Price	8
Charger properties	8
Fetch a single charger by ID	17
List multiple chargers	19
Session Entity	29
Role	29
Session properties	30
Fetch a single session by ID	34
List multiple Sessions	34
Create a session (start charging)	40
Stop a session (stop charging)	42
(Completed) Transaction entity	43
Role	43
Transaction properties	44
Fetch a single completed transaction by ID	46
List multiple completed transactions	47
Company entity	49
Role	49
Company properties	49
Fetch a single company by ID	50
List multiple companies	51

API basics

The ChargeLab API is "REST-like", using HTTP GET and POST methods against specific endpoints.

URLs

The URLs in this document are prefixed with the variable {baseUrl}. For this version of the API you must substitute:

```
https://api.chargelab.io/core/v1
```

Authentication

Unless otherwise specified, all requests must be authenticated, using an Authorization header in the HTTP request:

```
Authorization: x-auth {accessKey}
```

[Contact ChargeLab to have an access key generated for you.](#)

Payloads

All payloads (data) passed in requests and responses are in JSON format.

HTTP return codes

All operations return appropriate HTTP status codes:

- **200 OK** for successful operations.
- **40x** for issues with your request. Please correct and retry the request.
- **50x** or other errors may indicate a problem with our servers. If the error continues, please contact ChargeLab.

DateTime formats

All timestamps are specified/returned in ISO-8601 format:

```
yyyy-mm-ddThh:mm:ss.fffzzzzzz
```

Where:

- `yyyy-mm-dd` is the numeric year, month, and day of month
- `hh:mm:ss` are hours, minutes, seconds
- `fff` are fractions of a second
- `zzzzzz` is the timezone offset (from GMT), specified as:
 - `Z` meaning the time is in GMT, or
 - `+hh:mm` or `-hh:mm`

Role

When listing primary entities, or operating (fetching, inserting, updating) on a single entity (Chargers, Sessions, completed Transactions, Companies), you may specify a role value to indicate which role you are in, either a default role (personal user), or authorized company user. In general, when role is set to "COMPANY", you're able to access entities belonging to any company(ies) you have access to, as an authorized company user; when role is not set, you're able to access entities as a personal user, meaning you can access historical data only for charging history of you personal user (i.e. sessions authenticated over API by you).

Filtering

When listing primary entities (Chargers, Sessions, completed Transactions) you may specify filters to exclude all but the desired entries.

Filters are specified as URL query string parameters using one of the following formats:

- Field filter: `filter_{operation}[{fieldname}]=value`
- General filter: `filter_{operation}=value`

Where `{operation}` indicates how to use the supplied `value` to select entries. If the first format is used, the `{fieldname}` indicates the field to be checked.

Field filters are used in several places. Field filters can be any of:

- `eq` - equal
- `in` - equal to any (the value must be a comma-separated list)
- `lt` - less than
- `le` - less than or equal

- gt - greater than
- ge - greater than or equal
- like - contains the value anywhere in the string (not case-sensitive)

Exactly which field filters are valid for each field is indicated in the **Filter** column of the entities "Properties" table for the Charger, Session, or completed Transactions. If no filter is shown, you cannot filter on that field.

Some examples:

- Find the 3 chargers with the name "Charger 1", "Charger 2", and "Charger 3"

```
{baseUrl}/chargers?filter_in[name]=Charger1,Charger2,Charger3
```

- Find the chargers at locations with "lab" in the name (e.g. "Charge**Lab** Toronto", "Ottawa TEST**LAB**")

```
{baseUrl}/chargers?filter_like[location.name]=lab
```

Filters can also be combined (they are effectively "AND"ed—returned entries must match all of the filters). For example:

- Find all chargers between latitudes 42.1 and 43.9 degrees

```
{baseUrl}/chargers?filter_ge[location.latitude]=42.1&filter_le[location.latitude]=43.9
```

Pagination

When listing entities you do not have to fetch all that match your filters. You can indicate a subset of the results by specifying an `offset` (number of rows to skip) and `limit` (number of rows to return).

For example, assuming there are several thousand CompletedTransactions, and you have previously retrieved the first 200 of them, to retrieve the next "page" of 100 rows:

```
{baseUrl}/completedTransaction?offset=200&limit=100
```

If not specified, `offset` defaults to 0, and `limit` defaults to 1000.

Sample response:

```
{"totalCount":###, "entities":[...]}
```

"totalCount" is the number of all entity from the query result without offset and limit

"entities" is the list of entity after applying offset and limit on the full set of the result

Charger entity

Role

Chargers are associated with a company, and are either "public" or "private". They may also be marked "hidden". You may only access Chargers which belong to your company(ies), private Chargers which you have access to, or Chargers belonging to another company that are public and not hidden.

When listing Chargers...

- To access just those belonging to your company(ies), specify `role=COMPANY` as query parameter.
- To access just those public Chargers and private Chargers that you have access to as a driver, specify `role=DRIVER` as query parameter.
- To access Chargers belonging to your company(ies), as well as all public/visible Chargers from other companies, leave the query parameter `role` not specified.

When fetching a single Charger by ID...

- To search within Chargers belonging to your company(ies), specify `role=COMPANY` as query parameter.
- To search within those public Chargers and private Chargers that you have access to as a driver, specify `role=DRIVER` as query parameter.
- To search within Chargers belonging to your company(ies), as well as all public/visible Chargers from other companies, leave the query parameter `role` not specified.

Public access to charger information

In order to allow access to charger information when only the name of a charger is known, the following mechanism is implemented in our API.

Looking up a charger based on the charger name

The filter `filter_eq[name]` can be used to look up the information of a charger based on the charger name - which is visible to end-users unlike the internal charger identifier. The charger name appears on the charger sticker and also in the QR code URL.

```
GET {baseUrl}/chargers?filter_eq[name]=DAA-001
```

Limited information for private chargers

When listing charges with `filter_eq[name]`:

- For a charger that the user has access to, the result will contain all properties of the charger.
- For a charger that the user has no access to, the result will contain a limited set of properties (marked in the "Publicly accessible" column in the below Charger properties table).

Note that without `filter_eq[name]`, chargers are not included in the charger listing if the user does not have access to them.

For API calls without any authentication, the user is assumed to only have access to public chargers.

Price

The pricing fields indicate the cost to use the charger. Some chargers may be configured to have different costs depending on the time of use or the charging tier.

- The values shown under `currentPrice` are the current rates (those in effect at the time your request was issued).
- The values shown under `priceSchedule` describe how the rates vary over time or under different tiers.

Note: If no price rate is specified for a given time period, it means that the use of the charger is free during the period.

Charger properties

Property	Type	Publicly accessible	Filter	Description
chargerId	String		in	Unique ID of this Charger.
name	String	Yes	in, eq	The human-readable name/label of the Charger.
type	String	Yes		Indicates the type of Charger: • "Level 1" (AC - slow) • "Level 2" (AC) • "Level 3" (DC Fast Charger)
model	String			Human-readable model of the Charger (not in a

				specific format).
imageUrl	String			URL of an image showing the model of the Charger.
description	String			Description of the Charger. May be omitted or blank.
location.locationId	String		in, eq	Unique ID of the Location.
location.name	String		like	The human-readable name/label of the Location.
location.latitude	Double		ge, gt, le, lt	Geographic coordinates - latitude. Expressed as decimal degrees.
location.longitude	Double		ge, gt, le, lt	Geographic coordinates - longitude. Expressed as decimal degrees.
location.isoCountryCode	String			2-character ISO country code (ISO 3166-1 alpha-2).
location.stateOrRegion	String			Optional. Province/state/region. May be specified in full or as a code recognized as part of a postal address.
location.city	String			Optional.
location.streetAddress	String			Optional.
location.zipOrPostalCode	String			Optional.
location.timeZone	String			Time zone name e.g. "America/Toronto" or "America/Los_Angeles".
currentPrice.currency	String	Yes		The ISO-4217 currency code for the Charger pricing.

				Supported currency code include: • USD - US dollars • CAD - Canadian dollars
currentPrice.pricingType	String	Yes		Indicates how cost would be accrued during a charging session: • FREE • BILLED_BY_KWH • BILLED_BY_EFFECTIVE_CHARGING_TIME • BILLED_BY_TIME_PLUGGED_IN Note: Effective charging time is the amount of time taken to supply the delivered energy, if that energy was delivered at the maximum rate possible. "Effective" charging time is typically less than the actual charging time because the rate of energy transfer is typically less than the maximum.
currentPrice.unitPrice	Double	Yes		Current cost of charging based on pricing type. The unit is either per kWh or per hour
currentPrice.flatRate (deprecated)	Double	Yes		A fixed cost for a charging session (regardless of time or energy used). 0 or omitted if not applicable.
currentPrice.idleRate	Double	Yes		Idle Rate is for the session if the charging session is idle. The power consumed by charger is considerably low
currentPrice.flatFees	FlatFee[]	Yes		A list of 1 or more FlatFee - see below. Only included when flat fee exists
currentPrice.ratePerKilowatt Hour (deprecated)	Double	Yes		A cost per kWh of delivered energy. 0 or omitted if not applicable.
currentPrice.ratePerEffectiv eHour (deprecated)	Double	Yes		A cost per "effective" hour. Effective hours are the amount of time taken to supply the delivered energy, if that energy was delivered at

				the maximum rate possible. Note that "effective" hours are typically less than the actual charging time because the rate of energy transfer is typically less than the maximum possible. 0 or omitted if not applicable.
currentPrice.ratePerHourPluggedIn (deprecated)	Double	Yes		A cost per hour of charging (regardless of energy supplied during the period). 0 or omitted if not applicable.
priceBeforeAdjustment.unitPrice	Double	Yes		The unit price before the discount is applied.
priceBeforeAdjustment.idleRate	Double	Yes		The idle rate before the discount is applied
priceBeforeAdjustment.pricingType	String	Yes		The pricing type for the current charging session before the discount is applied.
priceSchedule.type	String	Yes		It indicates the schedule of the price: • FREE • PER_DAY • WEEKDAYS_WEEKEND • SAME Everyday • TIERED
priceSchedule.dayPrices	DayPrice[]	Yes		A list of 1 or more DayPrices - see below. Optional - Only included when scheduleType is not FREE or TIERED.
priceSchedule.tierPrices	TierPrice[]	Yes		A list of 1 or more TierPrices - see below. Optional - Only included when scheduleType is TIERED.

priceSchedule.idleRate	Double	Yes		The idle rate before the discount is applied. Only included when idle fee exists
priceSchedule.flatFees	FlatFee[]	Yes		A list of 1 or more FlatFee - see below. Only included when flat fee exists
owner.type	String		eq	Indicates the owner type of Charger • USER • COMPANY
owner.id	String		eq	Unique Id of the Charger owner
ports	Port[]	Yes		A list of 1 or more Ports - see below.
directions	String			Directions to the Charger location
contactInfo	String	Yes		Contact info of customer support
maintenanceFlag	Boolean	Yes		Indicates if the charger is under maintenance
whiteLabellingId	String			The internal id for the branding of the charger.
whiteLabellingName	String			The name of the branding of the charger.
userHasAccess	Boolean	Yes		Set to true if the user has access to the Charger. This Charger object contains all properties. Set to false if the user has no access to the Charger. This Charger object contains only the properties with a 'Yes' under the 'Publicly accessible' column in this table.

requirePaymentMethod	Boolean			Set to true if a charger requires a payment method (credit card, Apple Pay, etc) to start a charge, false otherwise.
----------------------	---------	--	--	--

Port

Property	Type	Description
portId	String	Unique ID of the port <i>within</i> the Charger.
connectorTypes	String[]	A list of connector types attached to this Port (usually there is only one). Valid connector types are: • CCS • Type 2 • Type 2 CCS • J1772 • CHAdeMO • TESLA • NACS • GB/T DC • GB/T AC
status	String	Current state of the Port: • AVAILABLE - ready and not being used • SESSION - currently in or starting a charging session • UNAVAILABLE - unavailable for use • UNKNOWN - unavailable for use or state cannot be determined Note: When the maintenance flag of a Charger is ON, all connectors will have the status of UNKNOWN.
statusWithoutMaintenance	String	Optional - only included if the maintenance flag for the Charger is set.

		Note: This property is intended only for specialized use by technicians. The Port status which should normally be reported to end users is in the "status" property. Current state of the Port as it would be reported if the maintenance flag were OFF. • AVAILABLE - ready and not being used • SESSION - currently in or starting a charging session • UNAVAILABLE - unavailable for use • UNKNOWN - unavailable for use or state cannot be determined
trailerAccessibility	String	Optional - only included if the trailer accessibility is indicated. • ACCESSIBLE - the port is accessible for vehicles with trailers • INACCESSIBLE - the port is not accessible for vehicles with trailers
expectedVoltage	double	Optional - only included if the value is available. The highest voltage observed from sampled charging sessions.
calculatedAmperage	double	Optional - only included if the value is available. The maximum amperage observed from sampled charging sessions. If the charger has not reported amperage for this port, the value is estimated using the model's maximum wattage and the voltage as reported during installation.

DayPrice

Property	Type	Description
day	String	MON, TUE, WED, THU, FRI, SAT, SUN, MON_FRI, SAT_SUN, DAILY
pricePeriods	PricePeriod[]	A list of 1 or more PricePeriods - see below. Note: If the pricing is FREE for a given time period, the array will not contain a PricePeriod entry for this time period. If the whole day is free, this array will be empty.

PricePeriod

Property	Type	Description
unitPrice	Double	Cost of charging in this period based on pricing type. The unit is either per kWh or per hour
currency	String	The ISO-4217 currency code for the pricing. Supported currency code include: • USD - US dollars • CAD - Canadian dollars
pricingType	String	Indicates how cost would be accrued during a charging session: • BILLED_BY_KWH • BILLED_BY_EFFECTIVE_CHARGING_TIME • BILLED_BY_TIME_PLUGGED_IN Note: Effective charging time is the amount of time taken to supply the delivered energy, if that energy was delivered at the maximum rate possible. "Effective" charging time is typically less than the actual charging time because the rate of energy transfer is typically less than the maximum.
startTime	String	Start time of this period. E.g. 17:00 Format: HH:mm 24H (Rounded to 15 minutes)
endTime	String	End time of this period. E.g. 17:45 Format: HH:mm 24H (Rounded to 15 minutes)
current	Boolean	If the price period is applicable at this moment of time, then True, else, it is False or omitted

TierPrice

Property	Type	Description
----------	------	-------------

unitPrice	Double	Cost of charging for the tier
currency	String	The ISO-4217 currency code for the pricing. Supported currency code include: • USD - US dollars • CAD - Canadian dollars
pricingType	String	Indicate how to price: • BILLED_BY_KWH • BILLED_BY_EFFECTIVE_CHARGING_TIME • BILLED_BY_TIME_PLUGGED_IN Note: Effective charging time is the amount of time taken to supply the delivered energy, if that energy was delivered at the maximum rate possible. "Effective" charging time is typically less than the actual charging time because the rate of energy transfer is typically less than the maximum.
duration	Double	• For BILLED_BY_EFFECTIVE_CHARGING_TIME and BILLED_BY_TIME_PLUGGED_IN, this will be the amount of time for which the tier applies • For BILLED_BY_KWH, this will be the amount of energy for which the tier applies
durationUnit	String	The unit of duration. One of below: • SECONDS • KILOWATTHOURS

FlatFee

Property	Type	Description
type	String	Name of flat fee: • SESSION_FEE
amount	Double	A fixed cost for a charging session (regardless of time or energy

used).

Fetch a single charger by ID

Method	GET
Endpoint	{baseUrl}/chargers/{chargerId}
Returns	A single Charger object
Possible HTTP 40x failures	403 Forbidden; if the caller is not linked to the company that owns the Charger, and the Charger is private or hidden 404 Not Found; if there is no Charger with the supplied {chargerId}

Example:

```
GET {baseUrl}/chargers/c2186d73-8303-49b2-ae2e-8e14c6836038
```

Response:

```
{
  "chargerId": "c2186d73-8303-49b2-ae2e-8e14c6836038",
  "name": "DAA-001",
  "type": "Level 2",
  "model": "Delta AC Mini",
  "location": {
    "locationId": "023a6d00-b400-4fd4-8809-9418b0a80992",
    "name": "ChargeLab Test",
    "latitude": 43.6351561,
    "longitude": -79.5301771,
    "isoCountryCode": "CA",
    "stateOrRegion": "ON",
    "city": "Toronto",
    "zipOrPostalCode": "M8Z 5B7",
    "streetAddress": "122 Judge Rd",
    "timeZone": "America/Toronto"
  },
  "ports": [
    {
```

```
    "portId": "1",
    "connectorTypes": [
      "J1772"
    ],
    "status": "AVAILABLE",
    "maxPowerKilowatts": 16,
    "trailerAccessibility": "ACCESSIBLE",
    "expectedVoltage": 240.0,
    "calculatedAmperage": 10.0,
  }
],
"imageUrl": "https://example.com/Delta+AC+Mini.jpg",
"description": "Private station for testing",
"currentPrice": {
  "currency": "CAD",
  "flatRate": 0.0,
  "ratePerKilowattHour": 0.0,
  "ratePerEffectiveHour": 1.0,
  "ratePerHourPluggedIn": 0.0,
  "pricingType": "BILLED_BY_EFFECTIVE_CHARGING_TIME",
  "unitPrice": 1.0,
  "flatFees": [
    {
      "type": "SESSION_FEE",
      "amount": 1.50
    }
  ]
},
"owner": {
  "type": "COMPANY",
  "id": "551a2ef5-346a-43e6-9d01-64d03f50e04a"
},
"Directions": "next to walmart store",
"contactInfo": "111-222-3333",
"maintenanceFlag": false,
"whiteLabellingId": "750a2ef5-346a-43e6-9d01-74d03f50e04c",
"whiteLabellingName": "Charger's brand name",
"userHasAccess": true,
"priceSchedule": {
  "Type": "SAME Everyday",
  "dayPrices": [
    {
```

```
    "day": "DAILY",
    "pricePeriods": [
      {
        "unitPrice": 1,
        "currency": "USD",
        "pricingType": "BILLED_BY_EFFECTIVE_CHARGING_TIME",
        "startTime": "00:00",
        "endTime": "24:00",
        "current": true
      }
    ]
  },
  "flatFees": [
    {
      "type": "SESSION_FEE",
      "amount": 1.50
    }
  ]
},
"requirePaymentMethod": true
}
```

List multiple chargers

Method	GET
Endpoint	{baseUrl}/chargers
Returns	A single PaginatedResult object, with a list of Chargers in the "entries" attribute
Possible HTTP 40x failures	400 Bad Request; if the specified query string parameters are not valid

Example 1:

List the first Charger of the company/companies you have access to.

```
GET {baseUrl}/chargers?role=COMPANY&limit=1
```

Response:

```
{
  "totalCount": 11,
  "entities": [
    {
      "chargerId": "c2186d73-8303-49b2-ae2e-8e14c6836038",
      "name": "DAA-001",
      "type": "Level 2",
      "model": "Delta AC Mini",
      "location": {
        "locationId": "023a6d00-b400-4fd4-8809-9418b0a80992",
        "name": "ChargeLab Test",
        "latitude": 43.6351561,
        "longitude": -79.5301771,
        "isoCountryCode": "CA",
        "stateOrRegion": "ON",
        "city": "Toronto",
        "zipOrPostalCode": "M8Z 5B7",
        "streetAddress": "122 Judge Rd",
        "timeZone": "America/Toronto"
      },
      "ports": [
        {
          "portId": "1",
          "connectorTypes": [
            "J1772"
          ],
          "status": "AVAILABLE",
          "maxPowerKilowatts": 16,
          "trailerAccessibility": "ACCESSIBLE",
          "expectedVoltage": 240.0,
          "calculatedAmperage": 10.0,
        }
      ],
      "imageUrl": "https://example.com/Delta+AC+Mini.jpg",
      "description": "Private station for testing",
      "currentPrice": {
        "currency": "CAD",
        "flatRate": 0.0,
        "ratePerKilowattHour": 0.0,
      }
    }
  ]
}
```

```
"ratePerEffectiveHour": 1.0,
"ratePerHourPluggedIn": 0.0,
"pricingType": "BILLED_BY_EFFECTIVE_CHARGING_TIME",
"unitPrice": 1.0,
"flatFees": [
  {
    "type": "SESSION_FEE",
    "amount": 1.50
  }
],
},
"owner": {
  "type": "COMPANY",
  "id": "551a2ef5-346a-43e6-9d01-64d03f50e04a"
},
"Directions": "next to walmart store",
"contactInfo": "111-222-3333",
"maintenanceFlag": false,
"whiteLabellingId": "750a2ef5-346a-43e6-9d01-74d03f50e04c",
"whiteLabellingName": "Charger's brand name",
"userHasAccess": true,
"priceSchedule": {
  "Type": "SAME EVERYDAY",
  "dayPrices": [
    {
      "day": "DAILY",
      "pricePeriods": [
        {
          "unitPrice": 1,
          "currency": "USD",
          "pricingType": "BILLED_BY_EFFECTIVE_CHARGING_TIME",
          "startTime": "00:00",
          "endTime": "24:00",
          "current": true
        }
      ]
    }
  ]
},
],
"flatFees": [
  {
    "type": "SESSION_FEE",
    "amount": 1.50
  }
]
```

```
    ],
    },
    "requirePaymentMethod": true
  }
]
}
```

Example 2:

List the first Charger owned by the specified company.

```
GET
{baseUrl}/chargers?filter_eq[companyId]=551a2ef5-346a-43e6-9d01-64d03f50e04a&limit=1
```

Response:

```
{
  "totalCount": 5,
  "entities": [
    {
      "chargerId": "c2186d73-8303-49b2-ae2e-8e14c6836038",
      "name": "DAA-001",
      "type": "Level 2",
      "model": "Delta AC Mini",
      "location": {
        "locationId": "023a6d00-b400-4fd4-8809-9418b0a80992",
        "name": "ChargeLab Test",
        "latitude": 43.6351561,
        "longitude": -79.5301771,
        "isoCountryCode": "CA",
        "stateOrRegion": "ON",
        "city": "Toronto",
        "zipOrPostalCode": "M8Z 5B7",
        "streetAddress": "122 Judge Rd",
        "timeZone": "America/Toronto"
      },
      "ports": [
        {
          "portId": "1",
          "connectorTypes": [
            "J1772"
          ],

```

```
        "status": "AVAILABLE",
        "maxPowerKilowatts": 16,
        "trailerAccessibility": "ACCESSIBLE",
        "expectedVoltage": 240.0,
        "calculatedAmperage": 10.0,
    }
],
"imageUrl": "https://example.com/Delta+AC+Mini.jpg",
"description": "Private station for testing",
"currentPrice": {
    "currency": "CAD",
    "flatRate": 0.0,
    "ratePerKilowattHour": 0.0,
    "ratePerEffectiveHour": 1.0,
    "ratePerHourPluggedIn": 0.0,
    "pricingType": "BILLED_BY_EFFECTIVE_CHARGING_TIME",
    "unitPrice": 1.0
},
"owner": {
    "type": "COMPANY",
    "id": "551a2ef5-346a-43e6-9d01-64d03f50e04a"
},
"Directions": "next to walmart store",
"contactInfo": "111-222-3333",
"maintenanceFlag": false,
"whiteLabellingId": "750a2ef5-346a-43e6-9d01-74d03f50e04c",
"whiteLabellingName": "Charger's brand name",
"userHasAccess": true,
"priceSchedule": {
    "Type": "SAME EVERYDAY",
    "dayPrices": [
        {
            "day": "DAILY",
            "pricePeriods": [
                {
                    "unitPrice": 1,
                    "currency": "USD",
                    "pricingType": "BILLED_BY_EFFECTIVE_CHARGING_TIME",
                    "startTime": "00:00",
                    "endTime": "24:00",
                    "current": true
                }
            ]
        }
    ]
}
```

```
        }
      ]
    },
    "requirePaymentMethod": true
  }
]
}
```

Example 3:

List the Chargers within specific geographic coordinates, and skip the first in the list.

GET

```
{baseUrl}/chargers?filter_ge[location.latitude]=43&filter_le[location.latitude]=44&filter_ge[location.longitude]=-80&filter_le[location.longitude]=-79&offset=1
```

Response:

```
{
  "totalCount": 2,
  "entities": [
    {
      "chargerId": "c2186d73-8303-49b2-ae2e-8e14c6836038",
      "name": "DAA-001",
      "type": "Level 2",
      "model": "Delta AC Mini",
      "location": {
        "locationId": "023a6d00-b400-4fd4-8809-9418b0a80992",
        "name": "ChargeLab Test",
        "latitude": 43.6351561,
        "longitude": -79.5301771,
        "isoCountryCode": "CA",
        "stateOrRegion": "ON",
        "city": "Toronto",
        "zipOrPostalCode": "M8Z 5B7",
        "streetAddress": "122 Judge Rd",
        "timeZone": "America/Toronto"
      },
      "ports": [
        {
          "portId": "1",
          "connectorTypes": [
            "J1772"
          ]
        }
      ]
    }
  ]
}
```

```
        "status": "AVAILABLE",
        "maxPowerKilowatts": 16,
        "trailerAccessibility": "ACCESSIBLE",
        "expectedVoltage": 240.0,
        "calculatedAmperage": 10.0,
    }
],
"imageUrl": "https://example.com/Delta+AC+Mini.jpg",
"description": "Private station for testing",
"currentPrice": {
    "currency": "CAD",
    "flatRate": 0.0,
    "ratePerKilowattHour": 0.0,
    "ratePerEffectiveHour": 1.0,
    "ratePerHourPluggedIn": 0.0,
    "pricingType": "BILLED_BY_EFFECTIVE_CHARGING_TIME",
    "unitPrice": 1.0
},
"owner": {
    "type": "COMPANY",
    "id": "551a2ef5-346a-43e6-9d01-64d03f50e04a"
},
"Directions": "next to walmart store",
"contactInfo": "111-222-3333",
"maintenanceFlag": false,
"whiteLabellingId": "750a2ef5-346a-43e6-9d01-74d03f50e04c",
"whiteLabellingName": "Charger's brand name",
"userHasAccess": true,
"priceSchedule": {
    "Type": "SAME EVERYDAY",
    "dayPrices": [
        {
            "day": "DAILY",
            "pricePeriods": [
                {
                    "unitPrice": 1,
                    "currency": "USD",
                    "pricingType": "BILLED_BY_EFFECTIVE_CHARGING_TIME",
                    "startTime": "00:00",
                    "endTime": "24:00",
                    "current": true
                }
            ]
        }
    ]
}
```

```
        }
      ],
    },
    "requirePaymentMethod": true
  }
]
```

Example 4:

Obtain the information for a charger via the charger name. In this example, the user has no access to the charger and thus only limited information is returned.

```
GET {baseUrl}/chargers?filter_eq[name]=DAA-001
```

Response:

```
{
  "totalCount": 1,
  "entities": [
    {
      "name": "DAA-001",
      "type": "Level 2",
      "ports": [
        {
          "portId": "1",
          "connectorTypes": [
            "J1772"
          ],
          "status": "AVAILABLE",
          "maxPowerKilowatts": 16,
          "trailerAccessibility": "ACCESSIBLE"
        }
      ],
      "currentPrice": {
        "currency": "CAD",
        "flatRate": 0.0,
        "ratePerKilowattHour": 0.0,
        "ratePerEffectiveHour": 1.0,
        "ratePerHourPluggedIn": 0.0,
        "pricingType": "BILLED_BY_EFFECTIVE_CHARGING_TIME",
        "unitPrice": 1.0
      },
      "contactInfo": "111-222-3333",
      "maintenanceFlag": false,
      "userHasAccess": false,
      "priceSchedule": {
        "Type": "SAME EVERYDAY",
        "dayPrices": [
          {
            "day": "DAILY",
            "pricePeriods": [
```

```
{
  {
    "unitPrice": 1,
    "currency": "USD",
    "pricingType": "BILLED_BY_EFFECTIVE_CHARGING_TIME",
    "startTime": "00:00",
    "endTime": "24:00",
    "current": true
  }
}
}
```

Session Entity

A Session is an attempted (possibly successful) charging session.

You may create a Session for Chargers which are free (no payment required) or paid (payment required) and are...

- accessible to you as a personal user (i.e., private Chargers where you are granted access and public and visible Chargers) or
- associated with a company that your user id/access key is associated with (i.e, it is one of your company's Chargers)

Sessions are created in "PREPARING", which means that the system is negotiating with the charger to start a charging session with an EV.

If successful, the status will change to "CHARGING", and remain there while charging. When charging is terminated (typically by the EV being unplugged, or a "Stop a Charging Session" operation being issued), the Session will move to status SUCCESSFUL. Successful Sessions will have the transactionId field populated; this can be used to retrieve the associated CompletedTransaction details.

If not successful, the status will move to FAILED, and the transactionId will not be populated. Instead, a reason for the failure will be supplied in the failureCode field.

In summary:

- PREPARING is the initial state, while the system attempts to start a transaction on the Charger.
- CHARGING means that a transaction has started and not yet finished.
- SUCCESSFUL means that the Session (and transaction) has ended, and an associated CompletedTransaction was created.
- FAILED indicates that an associated Transaction could not be created.

Once SUCCESSFUL or FAILED, no further updates/changes will be made to the Session.

Note that Transaction details will be available via the "completedTransactions" endpoint only when SUCCESSFUL status is reached.

Role

You may only access sessions you created, or sessions belonging to your company/companies.

When listing Sessions...

- To access Sessions belonging to your company/companies, specify `role=COMPANY` as query parameter.
- To access Sessions you created, leave the role unspecified or specify `role=DRIVER` as query parameter.

When fetching a single Session by ID...

- To search within the Sessions belonging to your company/companies, specify `role=COMPANY` as query parameter.
- To search within the Sessions you created, leave the role unspecified or specify `role=DRIVER` as query parameter.

When starting/stopping a Session...

- To start a new session or stop an active session as an authorized company user, specify `role=COMPANY` as query parameter.
- To start a new session or stop an active session as a personal user, leave the query parameter "role" unspecified.

Session properties

Property	Type	Filter	Description
sessionId	String	in	Unique ID of this Session.
chargerId	String	in, eq	ID of the associated Charger.
portId	String	in, eq	Port number (of the Charger).
createTime	String	gt, ge, lt, le	When the Session was created i.e., when. Formatted as an ISO date/time using the timezone of the Charger.
completeTime	String	gt, ge, lt, le	When this Session was completed (when it moved to SUCCESSFUL or FAILED status). Formatted as an ISO date/time using the timezone of the Charger. Will be null (missing) if the Session is still active.

status	String	in	One of: • PREPARING • CHARGING • SUCCESSFUL • FAILED
owner.type	String	eq	Indicates the owner type of Charger • USER • COMPANY
owner.id	String	eq	Unique Id of the charger owner
transactionId	String		ID of the associated Transaction, if any. Populated when the Session enters CHARGING status.
failureCode	String		A code describing the reason for failure when a Session enters the FAILED status: • STOPPED_BEFORE_TX - the Session was stopped before any associated Transaction was started (i.e. before charging could start). This may be caused by: ◦ a "stop charging" command was issued via the API, or ◦ the EV driver did not connect the EV within the allowed time, or ◦ some other issue preventing the charger from starting
startedVia	String		Indicates the initiator of the Session, one of the followings: • RFID • AUTOSTART • SYSTEM: the Session started remotely, such as via the app, site host dashboard, or a call to the start charging endpoint • PAYMENT_TERMINAL

			UNKNOWN
runningTotals			Running totals of the session will be populated only while the session status is CHARGING
runningTotals.cost	Double		Current cost for this active session Null if information not yet available
runningTotals.currency	String		The ISO-4217 currency code for the session. Supported currency code include: - USD - US dollars - CAD - Canadian dollars
runningTotals.costBreakdown	CostBreakdownItem[]		A list of 1 or more CostBreakdownItems - see below. Null if information not yet available
runningTotals.consumedEnergyKilowattHours	Double		Current energy delivered to EV Null if information not yet available
runningTotals.pluggedInTimeSeconds	Double		The cumulative amount of time during which the EV was connected to the port. Null if information not yet available
runningTotals.effectiveChargingTimeSeconds	Double		The charging time it would have taken to supply the delivered energy, if that energy was delivered at the maximum rate possible. "Effective" charging time is typically less than the actual charging time because the rate of energy transfer is typically less than the maximum. Null if information not yet available
idleFeeStatus	String		The current idle fee status - NOT_IMPOSED - Idle fee is either not a part of the pricing plan or not imposed - GRACE_PERIOD - The vehicle has entered an idle state but idle fees have not yet taken effect - IMPOSED - idle fee is taking effect

idleFeeStartTime	String		When the idle fee starts to take effect. Formatted as an ISO date/time using the timezone of the Charger.
stopInitiated	boolean		Set to true if 'Stop' is initiated for the session.

CostBreakdownItem

Property	Type	Description
periodStart	String	When this charging period started. Formatted as an ISO8601 date/time.
periodEnd	String	When this charging period ended. Formatted as an ISO8601 date/time.
periodCost	Double	Cost within this pricing window
currency	String	The ISO-4217 currency code for the period. Supported currency code include: • USD - US dollars • CAD - Canadian dollars
periodEnergyKilo wattHours	Double	KWh within this pricing window
pricingType	String	Pricing type for this pricing window: • BILLED_BY_KWH • BILLED_BY_EFFECTIVE_CHARGING_TIME • BILLED_BY_TIME_PLUGGED_IN • IDLE_RATE Note: Effective charging time is the amount of time taken to supply the delivered energy, if that energy was delivered at the maximum rate possible. "Effective" charging time is typically less than the actual charging time because the rate of energy transfer is typically less than the maximum.

unitPrice	Double	Unit price for this pricing window based on pricing type. The unit is per kWh or per hour for BILLED_BY_KWH, BILLED_BY_EFFECTIVE_CHARGING_TIME and BILLED_BY_TIME_PLUGGED_IN, and per minute for IDLE_RATE.
-----------	--------	---

Fetch a single session by ID

Method	GET
Endpoint	{baseUrl}/sessions/{sessionId}
Returns	a single Session object
Possible HTTP 40x failures	403 Forbidden; if the caller was not the creator of the Session when role is not set, or the caller doesn't have the access to the company owner of the Session when role is set to COMPANY 404 Not Found; if there is no Session with the supplied {sessionId}

Example:

```
GET {baseUrl}/sessions/1d37113f-d11b-41f9-9aec-8ffd2a4cd4c5
```

Response:

```
{
  "sessionId": "1d37113f-d11b-41f9-9aec-8ffd2a4cd4c5",
  "chargerId": "c2186d73-8303-49b2-ae2e-8e14c6836038",
  "portId": "1",
  "createTime": "2022-01-31T20:00:02.305104+03:00[Antarctica/Syowa]",
  "completeTime": "2022-02-02T00:04:46.141431+03:00[Antarctica/Syowa]",
  "status": "SUCCESSFUL",
  "owner": {
    "type": "COMPANY",
    "id": "551a2ef5-346a-43e6-9d01-64d03f50e04a"
  },
  "transactionId": "ca2c4435-cfe1-ba02-8587-fe0336c86c03",
  "startedVia": "SYSTEM"
}
```

List multiple Sessions

Method	GET
Endpoint	{baseUrl}/sessions
Returns	a single PaginatedResult object, with a list of Sessions in the "entries" attribute
Possible HTTP 40x failures	400 Bad Request; if the specified query string parameters are not valid

Example 1:

List all active Sessions (those in PREPARING or CHARGING state).

```
GET {baseUrl}/sessions?filter_in[status]=PREPARING,CHARGING
```

Response:

```
{
  "total": 3,
  "entities": [
    {
      "sessionId": "3973f717-9374-4871-bb6d-4aaa9f11db8a",
      "chargerId": "5433e1ac-9b44-418e-aed6-8802f6490079",
      "portId": "1",
      "createTime": "2023-06-20T16:39:46.11973-06:00[America/Toronto]",
      "status": "CHARGING",
      "owner": {
        "type": "COMPANY",
        "id": "1711cc42-a9a6-403b-b2a9-e241b887375f"
      },
      "transactionId": "9b641a42-1c82-4aa7-9f14-ebceb4e02f0c",
      "startedVia": "SYSTEM",
      "runningTotals": {
        "cost": 0.03125,
        "currency": "CAD",
        "costBreakdown": [
          {
            "periodStart":
              "2023-06-20T16:39:48.467-06:00[America/Toronto]",
```

```
        "periodEnd":
"2023-06-20T16:41:04.132-06:00[America/Toronto]",
        "periodCost": 0.03125,
        "currency": "CAD",
        "periodEnergyKilowattHours": 0.3,
        "pricingType": "BILLED_BY_EFFECTIVE_CHARGING_TIME",
        "unitPrice": 2
    }
],
    "consumedEnergyKilowattHours": 0.3,
    "effectiveChargingTimeSeconds": 56.25,
    "pluggedInTimeSeconds": 75.665
},
"idleFeeStatus": "NOT_IMPOSED"
},
{
    "sessionId": "e49d45ae-1926-4827-9c2c-9c6e8ec1909d",
    "chargerId": "21f56acc-5e1c-4c7a-b85b-6ea8b6c691a9",
    "portId": "1",
    "createTime": "2023-06-20T16:40:15.208532341-04:00[America/Toronto]",
    "status": "CHARGING",
    "owner": {
        "type": "COMPANY",
        "id": "1711cc42-a9a6-403b-b2a9-e241b887375f"
    },
    "transactionId": "27c48eb7-ded2-45d5-bf85-14c5d43f19ab",
    "startedVia": "SYSTEM",
    "runningTotals": {
        "cost": 0.29480704,
        "currency": "CAD",
        "costBreakdown": [
            {
                "periodStart":
"2023-06-20T16:40:24.387-04:00[America/Toronto]",
                "periodEnd":
"2025-06-20T16:46:40.387-04:00[America/Toronto]",
                "periodCost": 0.29480704,
                "currency": "CAD",
                "periodEnergyKilowattHours": 0.7370176,
                "pricingType": "BILLED_BY_EFFECTIVE_CHARGING_TIME",
                "unitPrice": 20.0
            }
        ]
    },
    ],
```

```
        "consumedEnergyKilowattHours": 0.737,
        "effectiveChargingTimeSeconds": 53.0652672,
        "pluggedInTimeSeconds": 540.114
    },
    "idleFeeStatus": "GRACE_PERIOD",
    "idleFeeStartTime": "2023-06-20T17:04:24.501-04:00[America/Toronto]"
},
{
    "sessionId": "68d39d9a-22d8-4b87-82c8-1cbc15cdae52",
    "chargerId": "0964d64a-e762-4224-8073-51a35e1c3b8b",
    "portId": "1",
    "createTime": "2023-06-20T16:39:48.467-06:00[America/Toronto]",
    "status": "PREPARING",
    "owner": {
        "type": "COMPANY",
        "id": "1711cc42-a9a6-403b-b2a9-e241b887375f"
    },
    "startedVia": "SYSTEM"
}
]
```

Example 2:

List all active Sessions belong to the company/companies you have access to.

```
GET {baseUrl}/sessions?role=COMPANY&filter_in[status]=PREPARING,CHARGING
```

Response:

```
{
  "totalCount": 3,
  "entities": [
    {
      "sessionId": "6794762b-36e5-484f-91ec-4eb1fb602f93",
      "chargerId": "875cfb24-6cb8-4e34-b09b-edf9b8629bb1",
      "portId": "1",
      "createTime": "2023-06-20T17:10:48.272528-04:00[America/Toronto]",
      "status": "CHARGING",
      "owner": {
        "type": "COMPANY",

```

```
    "id": "1711cc42-a9a6-403b-b2a9-e241b887375f"
  },
  "transactionId": "f1502035-c358-4618-b8d3-e94499f4a3f8",
  "startedVia": "SYSTEM",
  "runningTotals": {
    "cost": 2.621808333333334,
    "currency": "CAD",
    "costBreakdown": [
      {
        "periodStart":
"2023-06-20T17:10:50.655-04:00[America/Toronto]",
        "periodEnd":
"2023-06-20T17:26:34.515-04:00[America/Toronto]",
        "periodCost": 2.621808333333334,
        "currency": "CAD",
        "periodEnergyKilowattHours": 5.64,
        "pricingType": "BILLED_BY_EFFECTIVE_CHARGING_TIME",
        "unitPrice": 10.0
      }
    ],
    "consumedEnergyKilowattHours": 5.64,
    "effectiveChargingTimeSeconds": 943.8510000000001,
    "pluggedInTimeSeconds": 943.8600000000001
  },
  "idleFeeStatus": "NOT_IMPOSED"
},
{
  "sessionId": "e49d45ae-1926-4827-9c2c-9c6e8ec1909d",
  "chargerId": "21f56acc-5e1c-4c7a-b85b-6ea8b6c691a9",
  "portId": "1",
  "createTime": "2023-06-20T16:40:15.208532341-04:00[America/Toronto]",
  "status": "CHARGING",
  "owner": {
    "type": "COMPANY",
    "id": "1711cc42-a9a6-403b-b2a9-e241b887375f"
  },
  "transactionId": "27c48eb7-ded2-45d5-bf85-14c5d43f19ab",
  "startedVia": "SYSTEM",
  "runningTotals": {
    "cost": 0.29480704,
    "currency": "CAD",
    "costBreakdown": [
      {
```

```
        "periodStart":
"2023-06-20T16:40:24.387-04:00[America/Toronto]",
        "periodEnd":
"2025-06-20T16:46:40.387-04:00[America/Toronto]",
        "periodCost": 0.29480704,
        "currency": "CAD",
        "periodEnergyKilowattHours": 0.7370176,
        "pricingType": "BILLED_BY_EFFECTIVE_CHARGING_TIME",
        "unitPrice": 20.0
    },
    {
        "periodStart":
"2023-06-20T17:05:24.576-04:00[America/Toronto]",
        "periodEnd": "2023-06-20T17:17:24.62-04:00[America/Toronto]",
        "periodCost": 12.0,
        "currency": "CAD",
        "periodEnergyKilowattHours": 0.0,
        "pricingType": "IDLE_RATE",
        "unitPrice": 1.0
    }
],
"consumedEnergyKilowattHours": 0.737,
"effectiveChargingTimeSeconds": 53.0652672,
"pluggedInTimeSeconds": 540.114
},
"idleFeeStatus": "IMPOSED",
"idleFeeStartTime": "2023-06-20T17:04:24.501-04:00[America/Toronto]"
},
{
    "sessionId": "f1502035-c358-4618-b8d3-e94499f4a3f8",
    "chargerId": "afdd41f9-6e45-4aab-9914-4d168bf25d43",
    "portId": "1",
    "createTime": "2023-06-20T19:20:30.818-04:00[America/Toronto]",
    "status": "PREPARING",
    "owner": {
        "type": "COMPANY",
        "id": "1711cc42-a9a6-403b-b2a9-e241b887375f"
    },
    "startedVia": "SYSTEM"
},
{
    "sessionId": "827f7194-dc0d-4f71-a7fb-98cce283bb42",
    "chargerId": "0770d649-6be4-478c-a0ae-b32a6b539cd7",
```

```
"portId": "1",
"createTime": "2023-06-20T19:25:28.463379-04:00[America/Toronto]",
"status": "CHARGING",
"owner": {
  "type": "COMPANY",
  "id": "1711cc42-a9a6-403b-b2a9-e241b887375f"
},
"transactionId": "cd0ad2af-0d43-4db4-a505-dd4433bb1a94",
"startedVia": "SYSTEM",
"runningTotals": {
  "cost": 0.08344583333333334,
  "currency": "CAD",
  "costBreakdown": [
    {
      "periodStart":
"2023-06-20T19:25:30.818-04:00[America/Toronto]",
      "periodEnd":
"2023-06-20T19:26:30.899-04:00[America/Toronto]",
      "periodCost": 0.08344583333333334,
      "currency": "CAD",
      "periodEnergyKilowattHours": 0.12,
      "pricingType": "BILLED_BY_TIME_PLUGGED_IN",
      "unitPrice": 5.0
    }
  ],
  "consumedEnergyKilowattHours": 0.12,
  "effectiveChargingTimeSeconds": 22.5,
  "pluggedInTimeSeconds": 60.081
},
"idleFeeStatus": "NOT_IMPOSED"
}
]
```

Create a session (start charging)

Method	POST
Endpoint	{baseUrl}/sessions

Optional Parameters	paymentProcessing=[SELF, STRIPE (default)]
Payload	A JSON string with the following structure and attributes: <pre>{ "chargerId": "{chargerId}", "portId": "{portId}" }</pre>
Returns	The created Session object.
Possible HTTP 40x failures	400 Bad Request; if the supplied payload is not valid, or the charger or port do not exist, or the charger is not in an AVAILABLE state, or you do not have access to that charger, or it is an attempt to start a session on a paid charger without having a valid payment method associated with the user, or if you attempt to use payment self-processing without access.

Example:

```
POST {baseUrl}/sessions
```

Payload:

```
{
  "chargerId": "c2186d73-8303-49b2-ae2e-8e14c6836038",
  "portId": "1"
}
```

Response:

```
{
  "sessionId": "aafc42f5-4478-486f-9fa4-281428867f91",
  "chargerId": "c2186d73-8303-49b2-ae2e-8e14c6836038",
  "portId": "1",
  "createTime": "2022-02-02T23:08:03.742475-05:00[America/Toronto]",
  "status": "PREPARING",
  "owner": {
    "type": "COMPANY",
    "id": "551a2ef5-346a-43e6-9d01-64d03f50e04a"
  },
  "startedVia": "SYSTEM"
}
```

PaymentProcessing Example:

There are two options for payment processing.

STRIPE: the default if the paymentProcessing parameter is not set. ChargeLab will process the payment against the Stripe account attached to the account making the request.

SELF: specifies that the account making the request will be handling the payment on their end. The session totals will be calculated in the system as normal, but ChargeLab will not authorize or process the payment using Stripe. The SELF payment processing option is intended for roaming networks and third party applications that leverage their own payment processing, and is only accessible to partners with an approved API token - please speak to ChargeLab if you are interested in this option. Using the paymentProcessing=SELF without access will result in a 400 Bad Request.

```
POST {baseUrl}/sessions?paymentProcessing=SELF
```

Payload:

```
{
  "chargerId": "c2186d73-8303-49b2-ae2e-8e14c6836038",
  "portId": "1"
}
```

Response:

```
{
  "sessionId": "aafc42f5-4478-486f-9fa4-281428867f91",
  "chargerId": "c2186d73-8303-49b2-ae2e-8e14c6836038",
  "portId": "1",
  "createTime": "2022-02-02T23:08:03.742475-05:00[America/Toronto]",
  "status": "PREPARING",
  "owner": {
    "type": "COMPANY",
    "id": "551a2ef5-346a-43e6-9d01-64d03f50e04a"
  },
  "startedVia": "ROAM"
}
```

Stop a session (stop charging)

Method	POST
Endpoint	{baseUrl}/sessions/{sessionId}/stop
Payload	Any supplied payload is ignored
Returns	A Message object (the included message confirms the Session is stopped, but you do not need to check it; a normal 200 OK return code means the stop was processed successfully)
Possible HTTP 40x failures	400 Bad Request; if the Session is already in SUCCESSFUL or FAILED state 403 Forbidden; if the caller was not the creator of the Session when role is not set, or the caller doesn't have the access to the company owner of the Session when role is set to COMPANY 404 Not Found; if there is no Session with the supplied {sessionId}

Example:

```
POST {baseUrl}/sessions/aafc42f5-4478-486f-9fa4-281428867f91/stop
```

Payload:

```
{}
```

Response:

```
{  
  "message": "Session aafc42f5-4478-486f-9fa4-281428867f91 stopped"  
}
```

(Completed) Transaction entity

Transaction entities contain the details about energy used and amounts to be billed for a charging session. These are only available once the charging has ended.

Role

You may only access sessions you created, or sessions belonging to your company/companies.

When listing Transactions...

- To access transactions belonging to your company/companies, specify `role=COMPANY` as query parameter.
- To access transactions you created, leave the query parameter "role" unspecified.

When fetching a single Transaction by ID...

- To search within the Transactions belonging to your company/companies, specify `role=COMPANY` as query parameter.
- To search within the Transactions you created, leave the query parameter "role" unspecified.

Transaction properties

Property	Type	Filter	Description
transactionId	String	in	Unique ID of this Transaction.
portId	String		The ID of the port used for charging.
pluggedInTimeSeconds	Double		How long the car was physically connected to the port.
effectiveChargingTimeSeconds	Double		How long it would have taken to charge if energy had been delivered at the maximum rate allowable.
consumedEnergyJoules (<i>deprecated</i>)	Double		Energy delivered, in Joules.
consumedEnergyKilowattHours	Double		Energy delivered, in kWh.
startTime	String	gt, ge, lt, le	When charging started. Formatted as an ISO date/time.
stopTime	String	gt, ge, lt, le	When charging stopped. Formatted as an ISO date/time.

billedTotalAmount	Double		Cost of the charging session.
billedCurrency	String		The ISO-4217 currency code for billedAmount. Supported currency code include: • USD - US dollars • CAD - Canadian dollars
charger.chargerId	String	in, eq	The ID the Charger used for charging.
charger.name	String		The human-readable name/label of the Charger.
charger.location.locationId	String		Unique ID of the Location.
charger.location.name	String		The human-readable name/label of the Location.
charger.location.latitude	Double		Geographic coordinates - latitude. Expressed as decimal degrees.
charger.location.longitude	Double		Geographic coordinates - longitude. Expressed as decimal degrees.
charger.location.isoCountryCode	String		2-character ISO country code (ISO 3166-1 alpha-2).
charger.location.stateOrRegion	String		Optional. Province/state/region. May be specified in full or as a code recognized as part of a postal address.
charger.location.city	String		Optional.
charger.location.zipOrPostalCode	String		Optional.
charger.location.streetAddress	String		Optional.

charger.location.timeZone	String		Optional Time zone name e.g. "America/Toronto" or "America/Los_Angeles."
charger.owner.type	String		Indicates the owner type of Charger • USER • COMPANY • NETWORK
charger.owner.id	String		Unique Id of the Charger owner.
receiptId	String		Unique Id of the receipt.

Fetch a single completed transaction by ID

Method	GET
Endpoint	{baseUrl}/completedTransactions/{transactionId}
Returns	A single Transaction object
Possible HTTP 40x failures	403 Forbidden; if the caller was not the creator of the Transaction when role is not set, or the caller doesn't have the access to the company owner of the Transaction when role is set to COMPANY 404 Not Found; if there is no Session with the supplied {sessionId}

Example:

```
GET {baseUrl}/completedTransactions/118425d9-edd0-2118-81f7-77c60a5a218a
```

Response:

```
{
  "charger": {
    "chargerId": "c2186d73-8303-49b2-ae2e-8e14c6836038",
    "name": "DAA-001",
```

```
    "location": {
      "isoCountryCode": "CA",
      "stateOrRegion": "ON",
      "timeZone": "America/Toronto",
      "locationId": "023a6d00-b400-4fd4-8809-9418b0a80992",
      "name": "ChargeLab Test",
      "latitude": 43.6351561,
      "longitude": -79.5301771,
      "zipOrPostalCode": "M8Z 5B7",
      "streetAddress": "122 Judge Rd",
      "city": "Toronto"
    },
    "portId": "1",
    "pluggedInTimeSeconds": 631.5359999999998,
    "effectiveChargingTimeSeconds": 620.0,
    "consumedEnergyJoules": 4464000.0,
    "consumedEnergyKilowattHours": 1.24,
    "startTime": "2022-02-02T23:08:04.402-05:00",
    "stopTime": "2022-02-02T23:18:35.938-05:00",
    "billedTotalAmount": 0.0,
    "billedCurrency": "CAD",
    "transactionId": "118425d9-edd0-2118-81f7-77c60a5a218a",
    "owner": {
      "type": "COMPANY",
      "id": "551a2ef5-346a-43e6-9d01-64d03f50e04a"
    },
    "receiptId": "d67635fe-de91-4428-8729-33739c9046db"
  }
```

List multiple completed transactions

Method	GET
Endpoint	{baseUrl}/completedTransactions
Returns	A single PaginatedResult object, with a list of completed Transactions in the "entries" attribute
Possible HTTP 40x failures	400 Bad Request; if the specified query string parameters are not valid

Example:

List the first completed Transactions for a specific Charger.

```
GET
{baseUrl}/completedTransactions?role=COMPANY&filter_eq[charger.chargerId]=c2186d73-83
03-49b2-ae2e-8e14c6836038&limit=1
```

Response:

```
{
  "totalCount": 14,
  "entities": [
    {
      "charger": {
        "chargerId": "c2186d73-8303-49b2-ae2e-8e14c6836038",
        "name": "DAA-001",
        "location": {
          "isoCountryCode": "CA",
          "stateOrRegion": "ON",
          "timeZone": "America/Toronto",
          "locationId": "023a6d00-b400-4fd4-8809-9418b0a80992",
          "name": "ChargeLab Test",
          "latitude": 43.6351561,
          "longitude": -79.5301771,
          "zipOrPostalCode": "M8Z 5B7",
          "streetAddress": "122 Judge Rd",
          "city": "Toronto"
        }
      },
      "portId": "1",
      "pluggedInTimeSeconds": 1064.7039999999997,
      "effectiveChargingTimeSeconds": 880.0,
      "consumedEnergyJoules": 6336000.0,
      "consumedEnergyKilowattHours": 1.76,
      "startTime": "2022-01-29T22:04:10.483-05:00",
      "stopTime": "2022-01-29T22:21:55.187-05:00",
      "billedTotalAmount": 0.0,
      "billedCurrency": "CAD",
      "transactionId": "118425d9-edd0-2118-81f7-77c60a5a218a",
      "owner": {
        "type": "COMPANY",
        "id": "551a2ef5-346a-43e6-9d01-64d03f50e04a"
      },
      "receiptId": "d67635fe-de91-4428-8729-33739c9046db"
    }
  ]
}
```

```
]
}
```

Company entity

Company entities contain the details about companies the user has access to.

Role

When listing companies or fetching a single Company by ID, the user has to specify `role=COMPANY` as query parameter.

Company properties

Property	Type	Filter	Description
id	String		Unique ID of this Company.
name	String	eq	The human-readable name/label of the Company.
address.isoCountryCode	String		Optional. 2-character ISO country code (ISO 3166-1 alpha-2).
address.stateOrRegion	String		Optional. Province/state/region. May be specified in full or as a code recognized as part of a postal address.
address.city	String		Optional.
address.streetAddress	String		Optional.
address.zipOrPostalCode	String		Optional.

contact.name	String		Optional contact person name.
contact.email	String		Optional contact person's email.
contact.phone	String		Optional contact person's phone.
currency	String		The ISO-4217 currency code for billedAmount. Supported currency code include: • USD - US dollars • CAD - Canadian dollars

Fetch a single company by ID

Method	GET
Endpoint	{baseUrl}/companies/{companyId}?role=COMPANY
Returns	A single Company object
Possible HTTP 40x failures	403 Forbidden; if the caller doesn't have access to the company 404 Not Found; if there is no Company with the supplied {companyId}

Example:

```
GET {baseUrl}/companies/551a2ef5-346a-43e6-9d01-64d03f50e04a?role=COMPANY
```

Response:

```
{
  "id": "551a2ef5-346a-43e6-9d01-64d03f50e04a",
  "name": "ChargeLab Test",
  "address": {
    "isoCountryCode": "CA",
    "stateOrRegion": "ON",
```

```
{
  "city": "Toronto",
  "streetAddress": "122 Judge Rd",
  "zipOrPostalCode": "M8Z 5B7"
},
"contact": {
  "email": "test@example.com",
  "phone": "+14165551000"
},
"currency": "CAD"
}
```

List multiple companies

Method	GET
Endpoint	{baseUrl}/companies?role=COMPANY
Returns	A single PaginatedResult object, with a list of Companies in the "entities" attribute. If the user has no access to any company, the "entities" attribute is an empty list
Possible HTTP 40x failures	400 Bad Request; if the specified query string parameters are not valid

Example:

```
GET {baseUrl}/companies?role=COMPANY
```

Response:

```
{
  "totalCount": 2,
  "entities": [
    {
      "id": "0c015ab3-9024-4ceb-a2e5-f815fe23bf30",
      "name": "APITestCompany",
      "address": {
        "isoCountryCode": "CA",
        "stateOrRegion": "ON",
        "city": "Test Town",
        "streetAddress": "1 Testers Lane",
        "zipOrPostalCode": "A1A1A1"
      }
    }
  ]
}
```

```
    },
    "currency": "CAD"
  },
  {
    "id": "551a2ef5-346a-43e6-9d01-64d03f50e04a",
    "name": "ChargeLab Test",
    "address": {
      "isoCountryCode": "CA",
      "stateOrRegion": "ON",
      "city": "Toronto",
      "streetAddress": "122 Judge Rd",
      "zipOrPostalCode": "M8Z 5B7"
    },
    "contact": {
      "email": "test@example.com",
      "phone": "+14165551000"
    },
    "currency": "CAD"
  }
]
```