Sycada API

Sycada Technical Documentation

API Version: 1.2
Document Version: 1.2
Authors: George Tourkas (tourkas@sycada.com) , Rogier Mulder (mulder@sycada.com)

Authentication and Authorization

Authentication is based on the OpenID Connect protocol. Practically this means that clients need to authenticate themselves against an authentication server (https://auth.sycada.com) using OAuth2. Successful authentication will grant the client an encrypted token (called access token) with an associated expiration.This token can be further used to access resources on the resources server (https://api.sycada.com). After expiration, a new access token needs to be issued by authenticating again. An expired on invalid token will result in a 401 HTTP status from the resource server.

To get an authentication token the client must issue an HTTP POST request with content-type application/x-www-form-urlencoded to https://auth.sycada.com/oidc/token. The following shows how such a request will look with cURL.

```
curl --request POST \
  --url https://auth.sycada.com/oidc/token \
  --header 'content-type: application/x-www-form-urlencoded' \
  --data 'client_id=my_id&client_secret=my_secret&grant_type=client_credentials'
```

If the authentication succeeds, the HTTP status will be 200 and the payload a JSON object with the token type, the access token itself and its expiration in seconds:

```
{
  "token_type": "Bearer",
  "access_token": "CfDJ8L...l7xQ",
  "expires_in": 3600
}
```

If the authentication fails, the returned HTTP status will be 400 and the payload a JSON object with the error description:

```
{
  "error": "invalid_client"
}
```

Subscribing

Subscribing, means requesting the resources server to start pushing data to a designated URL (called CallBack URL). Currently this is done via HTTP POST with the payload being JSON. Pushing occurs in soft-real-time, meaning that the data will be pushed shortly (usually less than 1 second) after they are received and processed by the Sycada platform. The following list summarizes the currently pushed data entities.

Entities

Traces

A trace is a single data point created by a tracking device. A trace has the following properties:

Property	Description	Type	Unit	Availability
vehicle	The vehicle this trace comes from	Vehicle	n/a	Always
driver	The vehicle driver, if any.	Driver	n/a	If the vehicle is associated with a driver at the timestamp of the trace.
speed	The vehicle speed	Positive Integer	Level 2 Distance Unit (e.g. km, mi) per Hour	Always
heading	The vehicle heading	Integer in [0,359]	Radial degrees	Always
timestamp	The timestamp the trace has been sent from the tracking device.	Timestamp	n/a	Always
latitude	The WGS84 (GPS) latitude.	Decimal in [-90,90]	n/a	Always
longitude	The WGS84 (GPS) longitude.	Decimal in [-180,180]	n/a	Always
locations	The locations that the vehicle is into based on the latitude and longitude.	Array of Location.	n/a	If the organization account is set up accordingly.
battery_soc	The drive battery State-of-Charge	Decimal in [0-100]	%	Depends on device and vehicle type.
estimated_driving_range	The estimated driving range	Positive Decimal	Level 2 Distance Unit (e.g. km, mi)	Depends on device and vehicle type.

A sample trace in JSON looks like the following:

```
{
  "vehicle": {
     "id": "Renault Zoe",
     "license_plate": "AB-CD-12"
   },
  "driver": {
     "id": "JS101",
     "first_name": "John",
     "last_name": "Smith"
```

```
  },
  "speed": 85,
  "heading": 122,
  "timestamp": "2016-09-11T14:20:47",
  "latitude": 52.029296,
  "longitude": 5.639786,
  "locations": [
    {
      "id": "HQ",
      "name": "Head Quarters at Zaandam"
    },
    {
      "id": "Zaandam",
      "name": "Zaandam Area"
    }
  ],
  "battery_soc": 75.10,
  "estimated_driving_range": 210
}
```

Events

An event is created by when a trace matches certain criteria defined in Rules. An event has the following properties:

Property	Description	Type	Unit	Availability
vehicle	The vehicle this event comes from	Vehicle	n/a	Always
driver	The vehicle driver, if any.	Driver	n/a	If the vehicle is associated with a driver at the event timestamp.
speed	The vehicle speed	Positive Integer	Level 2 Distance Unit (e.g. km, mi) per Hour	Always
heading	The vehicle heading	Integer in [0,359]	Radial degrees	Always
timestamp	The timestamp the trace has been sent from the tracking device.	Timestamp	n/a	Always
latitude	The WGS84 (GPS) latitude.	Decimal in [-90,90]	n/a	Always
longitude	The WGS84 (GPS) longitude.	Decimal in [-180,180]	n/a	Always
description	The event description	String	n/a	Always

A sample trace in JSON looks like the following:

```
{
```

```
 "vehicle": {
    "id": "Renault Zoe",
    "license_plate": "AB-CD-12",
 },
 "driver": {
    "id": "JS101",
    "first_name": "John",
    "last_name": "Smith",
 },
 "speed": 85,
 "heading": 122,
 "timestamp": "2016-09-11T14:20:47",
 "latitude": 52.029296,
 "longitude": 5.639786,
 "description": "Renault Zoe started charging"
}
```


Trip Segments

A trip segment represents a vehicle movement from one location to another. A trip segment has the following properties.

Property	Description	Type	Unit	Availability
vehicle	The vehicle this trip comes from	Vehicle	n/a	Always
driver	The vehicle driver, if any.	Driver	n/a	If the vehicle is associated with a driver at least at the timestamp of the trip segment stop.
start	The start point of the trip	Point	n/a	Always
stop	The stop point of the trip	Point	n/a	Always
dist	Distance covered in the trip (on all energy modalities)	Integer	Level 1 Distance Unit (e.g. m, ft)	Always
drive_cycle	The drive cycle of the trip (e.g. Urban, Rural, Motorway etc)	Positive Integer	n/a	Always
scores				
energy_score	The energy score for the trip.	Score	n/a	Depends on device and vehicle type.
drivestyle_score	The main drive style score for the trip	Score	n/a	If defined.
other_scores	The collection of scores, other than the energy and the predefined drivestyle score.	Dictionary of of Score with the key being the score name.	n/a	If defined.
energy				

primary_energy	The primary energy reading	Energy	n/a	Depends on device and vehicle type.
duration				
dur	Duration of the trip: the time between the start and stop of the trip	Integer	s	Always
idling_dur	The time the vehicle was idling with the engine running.	Duration	n/a	Depends on device type and configuration.
driving_dur	The time the vehicle was driving.	Duration	n/a	Always
speed				
avg_speed	The trip average speed.	Positive Integer	Level 2 Distance Unit (e.g. km, mi) per Hour	Always
ineff_speed_dur	The driving time the speed was above the respective threshold during the trip.	Duration	n/a	Depends on device type.
excess_speed_dur	The driving time the speed was above the respective threshold during the trip.	Duration	n/a	Depends on device type.
dur_perc_per_speed_band	The percentage of driving time allocated in each speed band.	Array of 8 Decimals in [0-100]	n/a	Depends on device type.
acceleration				
ineff_accel_cnt	The counter of occurrences the acceleration was above the respective threshold during the trip.	Counter	n/a	Depends on device type.
hard_accel_cnt	The counter of occurrences the acceleration was above the respective threshold during the trip.	Counter	n/a	Depends on device type.
excess_accel_cnt	The counter of occurrences the acceleration was above the respective threshold during the trip.	Counter	n/a	Depends on device type.
ineff_decel_cnt	The counter of occurrences the deceleration was above the respective threshold during the trip.	Counter	n/a	Depends on device type.
hard_decel_cnt	The counter of occurrences the deceleration was above the respective threshold during the trip.	Counter	n/a	Depends on device type.
excess_decel_cnt	The counter of occurrences the deceleration was above the respective threshold during the trip.	Counter	n/a	Depends on device type.
fast_corner_cnt	The counter of occurrences the fast cornering was above the respective threshold during the trip.	Counter	n/a	Depends on device type.
dur_perc_per_accel_band	The percentage of driving time allocated in each acceleration band. The lower bands denote deceleration, whereas the higher ones denote acceleration.	Array of 8 Decimals in [0-100]	n/a	Depends on device type.

engine speed				
ineff_eng_speed_dur	The time the engine was in inefficient speed during the trip. Note that the percentage is of the driving time.	Duration	n/a	Depends on device type.
dur_perc_per_eng_speed_band	The percentage of driving time allocated in each engine speed band.	Array of 8 Decimals in [0-100]	n/a	Devices: ABC
engine load				
trip_time_engineload_per_band	The percentage of trip time allocated in each engine load band.	Array of 8 Decimals in [0-100]	n/a	Devices: ABC

Trip Segments / Counter

Property	Description	Type	Unit	Availability
value	The counter value	Positive Integer	n/a	Always
dist_unit	The distance unit to which the avg property is expressed to.	String Enum	Level 2 Distance Unit (e.g. km, mi)	Always
dist_num_of_units	The number of distance units (e.g. 10, 100 etc.) the avg property is expressed to.	Positive Integer	n/a	Always
avg	The averaged value.	Positive Decimal	/ (dist_unit X dist_num_of_units)	Always

Trip Segments / Point

Property	Description	Type	Unit	Availability
timestamp	The timestamp the point	Timestamp	n/a	Always
latitude	The WGS84 (GPS) latitude of the point	Decimal in [-90,90]	n/a	Always
longitude	The WGS84 (GPS) longitude of the point	Decimal in [-180,180]	n/a	Always
address	The mail address corresponding to the point's latitude and longitude.	String	n/a	Always, unless in the rare occasion the geocoding provider cannot find an address.
total_dist	The cumulated trip distance at that point. Depending on vehicle/tracker configuration this can be the vehicle odometer reading.	Integer	m	Always

Trip Segments / Energy

Property	Description	Type	Unit	Availability
type	The type of energy source.	String Enum	‘Diesel’ , ‘Gasoline’ , ‘CNG’	Always
unit	The energy unit. Depends on type.	String Enum	Level 1 Energy Unit (e.g. ‘l’, ‘kg’)	Always
dist_unit	The distance unit to which averaging (e.g. efficiency, savings etc.) properties are expressed to.	String Enum	Level 2 Distance Unit (e.g. km, mi)	Always
dist_num_of_units	The number of distance units (e.g. 10, 100 etc.) to which averaging (e.g. efficiency, savings etc.) properties are expressed to.	Positive Integer	n/a	Always
used	The actual energy use.	Positive Decimal	unit	Always
efficiency_used	The actual energy use efficiency.	Positive Decimal	unit per (dist_num_of_units X dist_unit)	Always
efficiency_target	The energy efficiency use target.	Positive Decimal	unit per (dist_num_of_units X dist_unit)	Always
efficiency_baseline	The energy efficiency use baseline.	Positive Decimal	unit per (dist_num_of_units X dist_unit)	Always
saved	Saved energy volume against baseline.	Positive Decimal	unit / (dist_num_of_units X dist_unit)	Always
saved_perc	Saved energy volume against baseline as a percentage of the baseline.	Decimal in [0-100]	n/a	Always
co2_emission	CO_2 emission	Decimal	Level 1 CO_2 Emission Unit (e.g. ‘kg’)	Always

Trip Segments / Duration

Property	Description	Type	Unit	Availability
value	The duration value	Positive Integer	s	Always
perc	This duration as a percentage of the total trip segment duration.	Integer in [0-100]	n/a	Always

Trip Segments / Score

Property	Description	Type	Unit	Availability
letter	The score letter	A-F	n/a	Always
value	The actual score value.	Integer in [0-100]	n/a	Always

A sample trip segment in JSON looks like the following:


```
{
 "vehicle": {
    "id": "Renault Zoe",
    "license_plate": "AB-CD-12"
 },
 "driver": {
    "id": "JS101",
    "first_name": "John",
    "last_name": "Smith"
 },
 "start": {
  "timestamp": "2016-09-12T15:11:02",
  "latitude": 52.439152,
  "longitude": 4.813642,
  "address": "Fluitekruidweg 8, NL-1508 AJ Zaandam",
  "total_dist": 469632755
 },
 "stop": {
  "timestamp": "2016-09-12T15:55:37",
  "latitude": 52.459583,
  "longitude": 4.830256,
  "address": "Provincialeweg 33A, NL-1506 MA Zaandam",
  "total_dist": 469636975
 },
 "dur": 12342,
 "distance": 24356,
 "drive_cycle": 0,
 "energy_score": {
  "letter": "A",
  "value": 95
 },
 "drivestyle_score": {
  "letter": "B",
  "value": 85
 },
 "other_scores": [
  {
   "breaking score": {
    "letter": "C",
    "value": 72
   }
  },
  {
   "cornerning score" : {
    "letter": "D"
    ,"value": 64
   }
  }
 ]
```

```
}
```

Vehicle

Property	Description	Type	Unit	Availability
id	A unique vehicle identifier within an organization.	String	n/a	Always
license_plate	The vehicle's license plate.	String	n/a	Always

Driver

Property	Description	Type	Unit	Availability
id	An optional but unique within the organization driver identifier (e.g. personnel number).	String	n/a	If set for the driver.
first_name	The driver's first name.	String	n/a	Always
last_name	The driver's last name.	String	n/a	Always

Locations

Property	Description	Type	Unit	Availability
id	An optional but unique within the organization location identifier.	String	n/a	If set for the location.
name	The vehicle's name.	String	n/a	Always

To subscribe, a client must issue an HTTP POST request to https://api.sycada.com/v1/subscription specifying a number of Stream Configurations. A Stream Configuration has a Stream Type and an CallBack URL. The valid Stream Types are:

- traces
- events
- tripsegments

```
curl --request POST \
```

```
 --url https://api.sycada.com/v1/subscription \
 --header 'authorization: Bearer CfDJ8L...l7xQ' \
 --header 'content-type: application/json' \
 --data '{ "stream_cfgs" : [ { "stream_type": "tripsegments", "callback_URL": "http://myhost.com:8080/tripsegments" } ] }'
```

If no subscription existed, an HTTP Status 201 will be returned along with a JSON object that holds the subscription expiration:

```
{
 "creation_timestamp_utc": "2016-09-11T15:01:54",
 "expiration_timestamp_utc": "2016-09-12T15:01:54",
 "stream_cfgs": [
  {
   "stream_type": "tripsegments",
   "callback_url": "http://myhost.com:8080/tripsegments"
  }
 ]
}
```

A client should re-subscribe before the expiration timestamp. Stream Configurations can be modified, if needed. A client can (re-)subscribe to any combination of entities as long as there is a single Callback URL for each Stream Type. An HTTP Status 200 will be returned when re-subscribing.

The listening side should return an HTTP Status 200 to acknowledge the data entity being processed. If any other status code is returned or a connection cannot be established the Sycada side will leave the data entity to its queue and will retry sending it after some seconds.

Data entities are also left in their queues after subscription expiration.

Unsubscribing

Unsubscribing, means requesting the resources server to stop pushing data.

To unsubscribe, a client must issue an HTTP DELETE request to https://api.sycada.com/v1/subscription :

```
curl --request DELETE \
 --url https://api.sycada.com/v1/subscription \
 --header 'authorization: Bearer CfDJ8L...l7xQ' \
 --header 'content-type: application/json' \
 --data '{ "permanently": false }'
```

A successful unsubscription will return a 200 HTTP Status. If a subscription does not exist, a 404 HTTP Status will be returned.

The client may optionally request the permanent deletion of the subscription. This means that all queued data will be removed. This is intended for use during development of the client to avoid possibly high volumes of accumulated data to be pushed.

Conventions

The current major version of the api is 1. Sycada may extend its API while the major version remains the same but it guarantees that compatibility with already implemented clients will not break. Compatibility across major versions is not guaranteed.

Both JSON input and output follow the snake case notation.

Units

All timestamps are in local timezone.

Measured	**Dependant**	**Level**	**Units**
Distance	Locality	1	Meters (m), Feet (f), etc
		2	Kilometers (km), Miles (mi) etc
Energy	Locality and Energy Modality	1	Liters (l), KiloWatt Hours (kWh), Kilograms (kg)
CO2 Emission	Locality	1	Kilograms (kg)

Client Credentials

These credentials are intended for machine-to-machine integration. The client needs to ensure that they remain private, meaning that they should not be included in the source or content of any web pages, posted to a developer or help forum, or disclosed to anybody outside the client's organisation.