



MULTIVERSO

MuVe – Client API

Software Design Documentation

The terms, conditions and contents of this document are confidential and can't be distributed or presented to entities that are not a part of the current project.

Version 051.01.01.01

01st October 2023

This page intentionally left blank

Contents

Authentication	6
API Functions.....	7
Get platform.....	7
Update platform.....	7
Get stations	8
Delete stations	9
Get station.....	9
Create station.....	10
Update station.....	10
Delete station.....	10
Get docks.....	11
Delete docks	11
Get dock	12
Create dock	12
Update dock	13
Delete dock	13
Get bikes.....	14
Set bikes status.....	14
Delete bikes.....	15
Get bike	16
Create bike	16
Update bike	16
Delete bike	17
Get trips.....	18
Get trip	19
Create/Update trip.....	19
Close trip	20
Get trip price	20
Get user history.....	21
Get users	22
Get user	22
Create user	23
Update user.....	24
Delete user	24

Get pricing tiers	25
Get cards	26
Get card	27
Create card	27
Update card	28
Delete card	28
Get alarms	29
Get events	30
Commands	31

This page intentionally left blank

Authentication

The API Authentication for the MuVe Bikessharing Platform is handled through HTTP Basic Authentication, as specified in RFC 7617. The username must be the PLATFORM_ID and the password must be the API_KEY, exactly as provided by your platform manager.

Example:

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** https://www.edmtech.pt/bikessharing/index.php/api/platform
- Authorization Tab:** Selected, showing 'Basic Auth' type.
- Username:** 0
- Password:** ad3c9edb286ea318707c1f6eb54ffe363123644a
- Show Password:** Checked (indicated by a checkbox).
- Params Tab:** Shows a note: "The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)"

All the requests must use the BASE URL as shown below:

BASE URL: https://bikessharing.soltrafego.pt/index.php/main/index

Important note: Some of the functions shown on the API overview document have been merged in other main endpoints or blocked from the outside for security reasons. As previously stated, the API overview is an internal document that presents the API, albeit these methods we reserve the right to block these requests from the outside for privacy/security reasons.

API Functions

GET

Get platform

Description: Gets the information table of the current platform

Endpoint: /api/platform

Parameters: None

Returns

```
{
  "success": true,
  "platform": {
    "id": "8",
    "name": "Alenquer",
    "city": "Alenquer",
    "address": "Complexo Municipal da Barnabé",
    "country": "Portugal",
    "logo": "",
    "num_users": "10",
    "num_bikes": "19"
  }
}
```

PUT

Update platform

Description: Updates the information table of the current platform

Endpoint: /api/platform

Parameters:

- name (string / optional)
- address (string / optional)
- city (string / optional)
- zipcode (string / optional)
- phone (string / optional)
- mobile (string / optional)
- email (string / optional)
- url (string / optional)
- trip_time_warning (int / optional) – Value in hours
- interval_between_trips (int / optional) – Value in seconds. Minimum time between two consecutive trips of the same user
- max_trip_time_before_interval_applies (int / optional) – Value in seconds. Trip time before interval_between_trips is applied

Returns

```
{
  "success": true
}
```

GET

Get stations

Description: Gets a list of stations for the current platform**Endpoint:** /api/stations**Parameters:** None**Returns**

```

{
  "success": true,
  "data": [
    {
      "id": "17",
      "name": "Terminal Rodoviario",
      "internal_id": "0008-P01",
      "address": "Largo Rainha Santa Isabel",
      "lat": "39.0558",
      "lon": "-9.0072",
      "status": "unlocked",
      "platform_id": "8",
      "debug_mode": "1",
      "auto_add_bikes": "0",
      "protocol": "edm-bsp",
      "order_number": "1",
      "active": "1",
      "created_on": "2019-07-23 22:48:48",
      "updated_on": "2020-01-29 12:46:21",
      "num_free": "5",
      "num_bikes": "0"
    },
    {
      "id": "18",
      "name": "Jardim Vaz Monteiro",
      "internal_id": "0008-P02",
      "address": "Jardim Vaz Monteiro",
      "lat": "39.0522",
      "lon": "-9.00879",
      "status": "unlocked",
      "platform_id": "8",
      "debug_mode": "1",
      "auto_add_bikes": "0",
      "protocol": "edm-bsp",
      "order_number": "2",
      "active": "1",
      "created_on": "2019-07-23 22:50:37",
      "updated_on": "2020-01-29 12:46:29",
      "num_free": "6",
      "num_bikes": "0"
    }
  ],
  "bikes": []
}

```


DELETE

Delete stations

Description: Deletes multiple station on the platform

Endpoint: /api/stations

Parameters: ids (string) – URL-encoded, comma-separated list of station IDs to delete

Returns

```
{
  "success": true
}
```

GET

Get station

Description: Gets the information from one station of the current platform

Endpoint: /api/station/ID

Parameters: None

Returns

```
{
  "success": true,
  "data": [
    {
      "id": "17",
      "name": "Terminal Rodoviario",
      "internal_id": "0008-P01",
      "address": "Largo Rainha Santa Isabel",
      "lat": "39.0558",
      "lon": "-9.0072",
      "status": "unlocked",
      "platform_id": "8",
      "debug_mode": "1",
      "auto_add_bikes": "0",
      "protocol": "edm-bsp",
      "order_number": "1",
      "active": "1",
      "created_on": "2019-07-23 22:48:48",
      "updated_on": "2020-01-29 12:46:21",
      "num_free": "5",
      "num_bikes": "0"
    }
  ]
}
```

POST

Create station

Description: Adds a new station to the current platform

Endpoint: /api/station

Parameters:

- name (string)
- internal_id (string)
- address (string / optional)
- lat (float / optional)
- lon (float / optional)
- order_number (int / optional)
- min_free_docks (int / optional)

Returns

```
{
  "success": true,
  "id": 1234
}
```

PUT

Update station

Description: Updates a station data

Endpoint: /api/station/ID

Parameters:

- name (string / optional)
- internal_id (string / optional)
- address (string / optional)
- lat (float / optional)
- lon (float / optional)
- order_number (int / optional)
- min_free_docks (int / optional)

Returns

```
{
  "success": true,
  "id": 1234
}
```

DELETE

Delete station

Description: Deletes a station on the platform

Endpoint: /api/station/ID

Parameters: None

Returns

```
{
  "success": true
}
```

GET

Get docks

Description: Gets a list of docks for the current platform**Endpoint:** /api/docks**Parameters:** station_id (string / optional)

Returns

```
{
  "success": true,
  "data": [
    {
      "id": "5",
      "internal_id": "5",
      "mvreports_id": "16282",
      "name": "Dock 5",
      "serial_number": "",
      "station_id": "1",
      "locked": "0",
      "previous_status": "unlocked",
      "has_bike": "0",
      "docked_bike_id": null,
      "active": "1",
      "created_on": "2017-03-30 12:18:51",
      "station_name": "1- Cais de São Roque"
    }
  ]
}
```

DELETE

Delete docks

Description: Deletes multiple station on the platform**Endpoint:** /api/docks**Parameters:** ids (string) – URL-encoded, comma-separated list of dock IDs to delete

Returns

```
{
  "success": true
}
```

GET

Get dock

Description: Gets the information from one dock**Endpoint:** /api/dock/ID**Parameters:** None

Returns

```
{
  "success": true,
  "data": [
    {
      "id": "5",
      "internal_id": "5",
      "mvreports_id": "16282",
      "name": "Dock 5",
      "serial_number": "",
      "station_id": "1",
      "locked": "0",
      "previous_status": "unlocked",
      "has_bike": "0",
      "docked_bike_id": null,
      "active": "1",
      "created_on": "2017-03-30 12:18:51",
      "station_name": "1- Cais de São Roque"
    }
  ]
}
```

POST

Create dock

Description: Adds a new dock to a station**Endpoint:** /api/dock

Parameters:

- name (string)
- id (string)
- station_id (int)

Returns

```
{
  "success": true,
  "id": 1234
}
```

PUT

Update dock

Description: Updates a dock data**Endpoint:** /api/dock/ID**Parameters:**

- name (string / optional)
- internal_id (string / optional)
-

Returns

```
{  
  "success": true  
}
```

DELETE

Delete dock

Description: Deletes a dock on the platform**Endpoint:** /api/dock/ID**Parameters:** None**Returns**

```
{  
  "success": true  
}
```

GET

Get bikes

Description: Gets a list of bikes for the current platform**Endpoint:** /api/bikes**Parameters:** None

Returns

```
{
  "success": true,
  "bikes": [
    {
      "id": "188532",
      "gps_id": "865067027414674",
      "name": "AE002",
      "internal_id": "b69cfac919",
      "lat": "39.0567",
      "lon": "-9.01262",
      "docked": "0"
    },
    {
      "id": "188533",
      "gps_id": "865067027050502",
      "name": "AE003",
      "internal_id": "dda2893fc9",
      "lat": "39.0566",
      "lon": "-9.01266",
      "docked": "0"
    }
  ]
}
```

PATCH

Set bikes status

Description: Sets the status for multiple bikes simultaneously**Endpoint:** /api/bikes

Parameters:

- ids (int array)
- status (string)

Returns

```
{
  "success": true
}
```

DELETE

Delete bikes

Description: Deletes multiple bikes on the platform

Endpoint: /api/bikes

Parameters: ids (string) – URL-encoded, comma-separated list of bike IDs to delete

Returns

```
{  
  "success": true  
}
```

GET

Get bike

Description: Gets the information from one bike**Endpoint:** /api/bike/ID**Parameters:** None

Returns

```
{
  "success": true,
  "data": {
    "id": "188783",
    "gps_id": "865067027420408",
    "name": "E9999",
    "internal_id": "fffffffffff",
    "lat": "40.4283",
    "lon": "-8.46315",
    "docked": "0"
  }
}
```

POST

Create bike

Description: Adds a new bike to a platform**Endpoint:** /api/bike

Parameters:

- name (string)
- id (string)
- gps_id (string)
- type (enum – can be “electrical” or “conventional”)

Returns

```
{
  "success": true,
  "id": 1234
}
```

PUT

Update bike

Description: Updates a bike data**Endpoint:** /api/bike/ID

Parameters:

- name (string / optional)

Returns

```
{
  "success": true
}
```


DELETE

Delete bike

Description: Deletes a bike on the platform**Endpoint:** /api/bike/ID**Parameters:** None**Returns**

```
{  
  "success": true  
}
```

GET

Get trips

Description: Gets a list of trips for the current platform**Endpoint:** /api/trips

Parameters:

- start (date / optional – YYYY-mm-dd format)
- end (date / optional – YYYY-mm-dd format)
- bike (int / optional – Bike ID)
- station (int / optional – Station ID)

Returns

```
{
  "success": true,
  "trips": [
    {
      "id": "3701",
      "user_id": "1204",
      "user_name": "a",
      "finished_on": "2020-02-05 12:54:23",
      "created_on": "2020-02-05 12:54:05",
      "price": "0",
      "payment_processed": "1",
      "status": "closed",
      "internal_id": "ee9cfbc049",
      "gps_id": null,
      "name": "ee9cfbc049 - Auto-created bike",
      "origin_name": "3- CONCRETA",
      "destination_name": "3- CONCRETA",
      "total_distance": "0"
    }
  ]
}
```

GET

Get trip

Description: Gets the information from one trip**Endpoint:** /api/trip/ID**Parameters:** None

Returns

```
{
  "success": true,
  "trip": {
    "id": "3701",
    "user_id": "1204",
    "user_name": "a",
    "finished_on": "2020-02-05 12:54:23",
    "created_on": "2020-02-05 12:54:05",
    "price": "0",
    "payment_processed": "1",
    "status": "closed",
    "internal_id": "ee9cfbc049",
    "gps_id": null,
    "name": "ee9cfbc049 - Auto-created bike",
    "origin_name": "3- CONCRETA",
    "destination_name": "3- CONCRETA",
    "total_distance": "0"
  }
}
```

POST

Create/Update trip

Description: Creates/updates a trip. Internal usage only. Not available for external integrations.**Endpoint:** /api/trip**Parameters:** N/A**Returns** N/A

PATCH

Close trip

Description: Closes a trip**Endpoint:** /api/trip/ID**Parameters:** None

Returns

```
{  
  "success": true  
}
```

GET

Get trip price

Description: Gets the price of a given trip**Endpoint:** /api/trip_price/TRIP_ID**Parameters:** None

Returns

```
{  
  "success": true,  
  "price": 10.0  
}
```

GET

Get user history

Description: Gets the history of a given user

Endpoint: /api/user_history/USER_ID

Parameters: None

Returns

```
{
  "success": true,
  "history": [
    {
      "id": "9427",
      "user_id": "1",
      "movement_type": "trip_ended",
      "platform_id": "6",
      "entity_id": "17362",
      "entity_name": "35a256e829",
      "bike_id": "188557",
      "bike_name": "STE016",
      "station_id": "6",
      "station_name": "P02- Fabrica de Santo Tirso",
      "value": "0",
      "created_on": "2019-12-04 15:54:59"
    }
  ]
}
```

GET

Get users

Description: Gets the list of users**Endpoint:** /api/users**Parameters:** - query (text / optional – An optional query to look for a user)

Returns

```
{
  "success": true,
  "users": [
    {
      "id": "1",
      "name": "José Silva ",
      "email": "teste@teste.pt",
      "token": "cd4ed3ab9262cba5f81a8b737b217c9669a3674b",
      "phone": "",
      "address": "",
      "nif": "",
      "platform_id": "1",
      "role": "admin",
      "first_login": "0"
    }
  ]
}
```

GET

Get user

Description: Gets a single user**Endpoint:** /api/user/USER_ID**Parameters:** None

Returns

```
{
  "success": true,
  "user": {
    "id": "1",
    "name": "José Silva ",
    "email": "teste@teste.pt",
    "token": "cd4ed3ab9262cba5f81a8b737b217c9669a3674b",
    "phone": "",
    "address": "",
    "nif": "",
    "platform_id": "1",
    "role": "admin",
    "first_login": "0"
  }
}
```

POST**Create user****Description:** Creates a user.**Endpoint:** /api/user

Parameters:

- name (text)
- email (text)
- password (text)
- phone (text)
- address (text)
- nif (text)
- id_card (text)
- birthday (date – yyyy-mm-dd)
- gender (male/female)
- zipcode (text)
- city (text)
- locality (text)
- ecname (text – Emergency contact)
- ecemail (text – Emergency contact)
- ecphone (text – Emergency contact)
- ecaddress (text – Emergency contact)
- ecnif (text – Emergency contact)

Returns

```
{  
  "success": true  
}
```

PUT

Update user

Description: Updates a user.

Endpoint: /api/user/USER_ID

Parameters:

- name (text)
- email (text)
- password (text)
- phone (text)
- address (text)
- nif (text)
- id_card (text)
- birthday (date – yyyy-mm-dd)
- gender (male/female)
- zipcode (text)
- city (text)
- locality (text)
- ecname (text – Emergency contact)
- ecemail (text – Emergency contact)
- ecphone (text – Emergency contact)
- ecaddress (text – Emergency contact)
- ecnif (text – Emergency contact)

Returns

```
{  
  "success": true  
}
```

DELETE

Delete user

Description: Deletes a user

Endpoint: /api/user/USER_ID

Parameters: None

Returns

```
{  
  "success": true  
}
```


GET

Get pricing tiers

Description: Gets the pricing tiers**Endpoint:** /api/tiers**Parameters:** None

Returns

```
{
  "success": true,
  "tiers": [
    {
      "id": "7",
      "platform_id": "1",
      "name": "Bikesharing 30 min",
      "active": "1",
      "created_on": "2019-09-16 02:35:00"
    }
  ]
}
```

GET

Get cards

Description: Gets the existing cards**Endpoint:** /api/cards**Parameters:** None

Returns

```
{
  "success": true,
  "cards": [
    {
      "id": "1439",
      "internal_id": "8a268397b8",
      "card_number": "165416515615",
      "user_id": "0",
      "type": "13",
      "plafond": "12",
      "valid_from": null,
      "valid_to": null,
      "default_card": "0",
      "maintenance_card": "1",
      "platform_id": "1",
      "active": "1",
      "updated_by": "1",
      "updated_on": "2019-09-23 14:16:07",
      "created_by": "1",
      "created_on": "2019-09-20 10:25:36",
      "deleted_by": "0",
      "deleted_on": null
    }
  ]
}
```

GET

Get card

Description: Gets the information from one card**Endpoint:** /api/card/ID**Parameters:** None

Returns

```
{
  "success": true,
  "card": {
    "id": "1439",
    "internal_id": "8a268397b8",
    "card_number": "165416515615",
    "user_id": "0",
    "type": "13",
    "plafond": "12",
    "valid_from": null,
    "valid_to": null,
    "default_card": "0",
    "maintenance_card": "1",
    "platform_id": "1",
    "active": "1",
    "updated_by": "1",
    "updated_on": "2019-09-23 14:16:07",
    "created_by": "1",
    "created_on": "2019-09-20 10:25:36",
    "deleted_by": "0",
    "deleted_on": null
  }
}
```

POST

Create card

Description: Adds a new card to a platform**Endpoint:** /api/card

Parameters:

- card_number (string)
- initial_plafond (float)
- user_id (string – can be zero)
- type (id – The type ID)
- maintenance_card (boolean – 1 or 0)
- virtual_card (boolean – 1 or 0)

Returns

```
{
  "success": true,
  "id": 1234
}
```

PUT

Update card

Description: Updates a card data**Endpoint:** /api/card/ID

Parameters:

- card_number (string)
- plafond (float)
- user_id (string – can be zero)
- type (id – The type ID)
- maintenance_card (boolean – 1 or 0)
- virtual_card (boolean – 1 or 0)

Returns

```
{
  "success": true
}
```

DELETE

Delete card

Description: Deletes a card on the platform**Endpoint:** /api/card/ID**Parameters:** None

Returns

```
{
  "success": true
}
```

GET Get alarms

Description: Gets the existing alarms

Endpoint: /api/alarms

Parameters: None

Response:

```
{
  "success": true,
  "alerts": [
    {
      "id": "297842",
      "platform_id": "1",
      "severity": "error",
      "code": "station_too_empty",
      "extra_data": "{ \"station\": { \"id\": \"1\", \"name\": \"1- Cais de S
        \\\n\\o Roque\\\", \"internal_id\": \"9999-
        P09\\\", \"address\": \"Cais Dos RemadoresOlimpicos\\\", \"lat\": \"40.6443\\\", \"lon\\
        \": \"-
        8.65592\\\", \"platform_id\": \"1\\\", \"status\": \"unlocked\\\", \"debug_mode\": \"0\\
        \", \"auto_add_bikes\": \"0\\\", \"protocol\": \"edm-
        bsp\\\", \"order_number\": \"0\\\", \"min_free_docks\": \"0\\\", \"max_free_docks\": \"0
        \\\", \"active\": \"1\\\", \"created_on\": \"2017-03-
        29 23:59:34\\\", \"updated_on\": \"2020-01-
        14 14:11:58\\\" }, \"docks\": \"0\\\", \"empty\": \"0\\\", \"full\": \"0\\\" }\",
      \"entity_id\": \"1\",
      \"dispatched\": \"1\",
      \"created_on\": \"2020-03-11 15:00:17\"
    }
  ]
}
```

GET

Get events

Description: Gets the existing events**Endpoint:** /api/events**Parameters:** None

Returns

```
{
  "success": true,
  "events": [
    {
      "id": "297842",
      "platform_id": "1",
      "severity": "error",
      "code": "station_too_empty",
      "extra_data": "{\"station\":{\"id\":\"1\",\"name\":\"1- Cais de São Roque\",\"internal_id\":\"9999-P09\",\"address\":\"Cais Dos Remadores Olimpicos\",\"lat\":\"40.6443\",\"lon\":\"-8.65592\",\"platform_id\":\"1\",\"status\":\"unlocked\",\"debug_mode\":\"0\",\"auto_add_bikes\":\"0\",\"protocol\":\"edmbsp\",\"order_number\":\"0\",\"min_free_docks\":\"0\",\"max_free_docks\":\"0\",\"active\":\"1\",\"created_on\":\"2017-03-29 23:59:34\",\"updated_on\":\"2020-01-14 14:11:58\"},\"docks\":{\"0\",\"empty\":\"0\",\"full\":\"0\"}}",
      "created_on": "2020-03-11 15:00:17"
    }
  ]
}
```

Commands

The commands shown below allow will be sent from the platform to the corresponding stations/docks, limited by the API permissions. All commands are logged.

Command	Method	Parameters
/main/query_station_status/STATION_ID	GET	
/main/query_dock_status/DOCK_ID	GET	
/main/release_bike/STATION_ID	POST	docks (int array)
/main/lock_station/STATION_ID	GET	
/main/unlock_station/STATION_ID	GET	
/main/lock_dock/STATION_ID	POST	docks (int array)
/main/unlock_dock/STATION_ID	POST	docks (int array)
/main/reset_dock/STATION_ID	POST	docks (int array)