

# API Clients: Basic Access (client\_credentials) - Downloadable version

Clients with grant type *client\_credentials* have a basic access to the API.

They will be able to make only two requests:

- Get all city stations
- Get all city stations availability

## NOTE:

In order to access the API, the developer must know the CLIENT\_ID and CLIENT\_SECRET key for their given client. For now, API developers team will take care of creating and providing these constant values to the client.

For more information, please check: [API Clients: Creating new clients](#)

- [First Step: API Access Token](#)
- [Second Step: Accessing to the API Methods](#)
  - [Get all city stations](#)
    - Request
    - Response
  - [Get all city stations availability](#)
    - Request
    - Response

## First Step: API Access Token

The only thing needed to access the API is a valid **access\_token**.

Each **access\_token** identifies an specific client.

Passing the **client\_id** and the **client\_secret** to the OAuth server it will respond with an **access\_token**.

This type of access\_token won't link an existing user with the client. Having this tokens won't let access to user specific information.

To get the access\_token following this grant type flow, we have to request the following url:

```
https://pubsbapi.smartbike.com/oauth/v2/token?client_id={CLIENT_ID}&client_secret={CLIENT_SECRET}&grant_type=client_credentials
```

This request will output:

```
{
  "access_token": "{ACCESS_TOKEN}",
  "expires_in": 3600,
  "refresh_token": "{REFRESH_TOKEN}",
  "scope": null,
  "token_type": "bearer"
}
```

- **NOTE:** This response comes with fields in snake\_case, instead of CamelCase, because it is part of OAuth standard and not part of the API itself. We leave them like this, because they will be used in the URIs in snake\_case as well as URI's are case insensitive

The access\_token will be valid for an hour (3600 seconds).

It can either create a new `access_token` whenever you want or refresh the `access_token` using the **refresh\_token** like this:

```
https://pubsbapi.smartbike.com/oauth/v2/token?client_id={CLIENT_ID}&client_secret={CLIENT_SECRET}&grant_type=refresh_token&refresh_token={REFRESH_TOKEN}
```

This request will output the same as above with a new `access_token` and `refresh_token`.

## Second Step: Accessing to the API Methods

In order to use the API it is needed to pass the **access\_token** on each call.

This can be done either by adding the **access\_token** to the query string or by adding an authorization bearer on the request header.

Added to the query string:

```
https://pubsbapi.smartbike.com/api/v1/api_section/api_method?access_token={ACCESS_TOKEN}
```

As an http header:

```
Authorization: Bearer {ACCESS_TOKEN}
```

With this grant type, the API client will be able to:

### Get all city stations

Get basic data from the API Client city related stations

#### Request

```
https://pubsbapi.smartbike.com/api/v1/stations.json?access_token={ACCESS_TOKEN}
```

#### Response

Successfully responses will have a 200 HTTP status code

For each station will be delivered:

Parameter	Type
id	integer

name	string
address	string
addressNumber	string
altitude	integer
districtCode	string
districtName	string
nearbyStations	array of nearby station id's (integers)
location	array
zipCode	string
stationType	string

• **NOTE:**

- The possible values for the field stationType are BIKE (normal bike), ELECTRIC\_BIKE (only for Barcelona), BIKE,TPV (For the Mexican stations that allow to buy a subscription) due to the future introduction of a new service for renting electric bikes.
- The field addressNumber might come with a non-numeric value for some cities, and because of that it has string type.
- The field altitude is the altitude over the sea level of the station and might come as null for some cities.

Location Array

Parameter	Type
lat	float
lon	float

The latitude and longitude are the coordinates of the stations based on World Geodetic System (WGS84)

### Response Example

```
{
  "stations": [
    {
      "id": 1,
      "name": "01 - C/ GRAN VIA CORTS CATALANES 760",
      "address": "Gran Via Corts Catalanes",
      "addressNumber": "760",
      "zipCode": "08013",
      "districtCode": "2",
      "districtName": "Eixample",
      "nearbyStations": [
        24,
        369,
        387,
        426
      ],
      "location": {
        "lat": 41.397952,
        "lon": 2.180042
      },
      "stationType": "BIKE",
    }
  ]
}
```

```
"altitude": 21
},
{
  "id": 2,
  "name": "02 - PL. TETUAN ,8-9",
  "address": "Plaza Tetuán",
  "addressNumber": "8",
  "zipCode": "08010",
  "districtCode": "2",
  "districtName": "Eixample",
  "nearbyStations": [
    360,
    368,
    387,
    414
  ],
  "location": {
    "lat": 41.394272,
    "lon": 2.175169
  },
  "stationType": "BIKE",
  "altitude": 21
},
{
  "id": 3,
  "name": "03 - C/ D'ALÍ BEI 44",
  "address": "Ali Bei",
  "addressNumber": "44",
  "zipCode": "08013",
  "districtCode": "2",
  "districtName": "Eixample",
  "nearbyStations": [
    4,
    6,
    119,
    419
  ],
  "location": {
    "lat": 41.393699,
    "lon": 2.181137
  },
  "stationType": "BIKE",
  "altitude": 21
},
{
  "id": 426,
  "name": "426 - C/ DE RIBES,59 B",
  "address": "Carrer de Ribes",
  "addressNumber": "59",
  "zipCode": "08033",
  "districtCode": "2",
  "districtName": "Eixample",
  "nearbyStations": [
```

```
1,  
24,  
119,  
372  
],  
"location": {  
  "lat": 41.398305,  
  "lon": 2.183023  
},  
"stationType": "BIKE",  
"altitude": 21  
},  
{  
  "id": 428,  
  "name": "428 - C/ PUJADES, 103",  
  "address": "Carrer Pujades",  
  "addressNumber": "103",  
  "zipCode": "08005",  
  "districtCode": "8",  
  "districtName": "Sant Martí",  
  "nearbyStations": [  
    142,  
    149,  
    153,  
    161  
  ],  
  "location": {  
    "lat": 41.398389,  
    "lon": 2.196261  
  },  
  "stationType": "BIKE",  
  "altitude": 11
```

```
}
]
}
```

## Get all city stations availability

Get availability from the API Client city related stations

### Request

```
https://pubsbapi.smartbike.com/api/v1/stations/status.json?access_token={ACCESS_TOKEN}
```

### Response

Successfully responses will have a 200 HTTP status code

For each station will be delivered:

Parameter	Type
id	integer
status	string (OPN for open stations, CLS for Closed stations)
availability	array

Availability Array:

Parameter	Type
bikes	integer
slots	integer

#### Response Example

```
{
  "stationsStatus": [
    {
      "id": 1,
      "status": "OPN",
      "availability": {
```

```
    "bikes": 3,  
    "slots": 0  
  },  
  {  
    "id": 2,  
    "status": "OPN",  
    "availability": {  
      "bikes": 0,  
      "slots": 12  
    }  
  },  
  {  
    "id": 3,  
    "status": "OPN",  
    "availability": {  
      "bikes": 1,  
      "slots": 32  
    }  
  },  
  {  
    "id": 4,  
    "status": "OPN",  
    "availability": {  
      "bikes": 0,  
      "slots": 8  
    }  
  },  
  {  
    "id": 5,  
    "status": "OPN",  
    "availability": {  
      "bikes": 0,  
      "slots": 10  
    }  
  },  
  {  
    "id": 6,  
    "status": "OPN",  
    "availability": {  
      "bikes": 0,  
      "slots": 14  
    }  
  },  
  {  
    "id": 7,  
    "status": "OPN",  
    "availability": {  
      "bikes": 1,  
      "slots": 19  
    }  
  },  
  {
```

```
"id": 8,
"status": "OPN",
"availability": {
  "bikes": 1,
  "slots": 5
}
},
{
  "id": 9,
  "status": "OPN",
  "availability": {
    "bikes": 0,
    "slots": 23
  }
},
{
  "id": 10,
  "status": "OPN",
  "availability": {
    "bikes": 1,
    "slots": 26
  }
},
{
  "id": 11223,
  "status": "OPN",
  "availability": {
    "bikes": 0,
    "slots": 0
  }
}
```

