



OneCenter APIs

Version: 1.1

Date: Milano, 14 February 2023



1	Introduzione	3
2	Autenticazione.....	3
2.1	OAUTH TOKEN	3
2.2	Successive chiamate OAUTH	5
3	API	5
3.1	Facilities	5
3.2	Facility Free Places.....	6
4	Utente XXXX	8



1 Introduzione

Il sistema OneCenter espone API per interrogare il parcheggio ed ottenere l'elenco dei posti liberi in formato JSON (Content-Type: application/json).

2 Autenticazione

Il sistema autentica ed autorizza i client per mezzo di un server OAUTH integrato. Per dialogare con il sistema è necessario ottenere un token di autorizzazione, e poi utilizzarlo nelle successive chiamate.

2.1 OAUTH TOKEN

Per ottenere un token OAUTH è necessario effettuare una chiamata all'endpoint:

POST <https://www.onecenter.info/oauth/token>



La chiamata deve passare in POST, form urlencoded, i parametri:

```
client_id: [identificatore del chiamante]
client_secret: [password chiamante]
grant_type: "password"
username: [login]
password: [password]
```

Per esempio:

POST ▼ <https://www.onecenter.info/oauth/token>

Params Authorization Headers (7) **Body** ● Pre-request Script Tests Settings

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

	KEY	VALUE
<input checked="" type="checkbox"/>	username	[REDACTED]
<input checked="" type="checkbox"/>	password	[REDACTED]
<input checked="" type="checkbox"/>	grant_type	password
<input checked="" type="checkbox"/>	client_id	[REDACTED]
<input checked="" type="checkbox"/>	client_secret	[REDACTED]

Se effettuata con le corrette credenziali, la chiamata restituirà il token in un tracciato JSON così composto:

```
{
  "access_token": "FYOHDC79-
MTzCwQihl8M0tDGweoSNKYIo7vzK7BQYyf1vPRTgm7gG3cFqwGSwz9s5ZQrEH0kzgylyEnc1QI16qb-
N_FVgOivxm8OCsXeu5YukoB54fvEITVaE7NTqaZ_TicrDIqktpQhV3hcOm-
KnWbqpk7FENrjxp4pWpJGeHduvGexWIysuxwCgtvRumOLjLAG_cfj0Vc9NXJ0St0ydzP7hC6cUQFHVQv
Di9o6ObuLxMjhJA_1BjoFMyAeoPQrIAyV1Kvx0nTTDOhzbyvmeQ-A0oajhzHpSXQm-
iShHHH1PvYP2HJjrrFur0RU9prLZhAuQe4TSwtTJkkmsznc6mu69OnuE7rUpOrEa81SYuptgejxeqETbE
yzgJ-
9terYaBVzaVvgVIBBbAgfviAI004cYSwoTGbh9KpVb4GnfObPrZdh3MFliCVSxlFUW7fDeU69qJ_4qMC2b
sb8tBqtAMvReQIupwRmDS0ZS6SC1hmMyiEfKHR1zybzEOqAuxq",
  "token_type": "bearer",
  "expires_in": 1799,
  "refresh_token": "jIBwIKUS1GhPSIe-_gX4FROF033YVGWQ8BvsP6_bRbpS47lQH-
```



```
W_QMZ8RrmtXw_wnyv7WKMgmClmmhoo4QVb_X_BW4DULNdZVuWBlEvxSunPBblvo8lZyZdOqU4MV
eYdBruzQcLRxLXxJ6rtJSzi8vTSziCmW65Sxo3vypxwFU35_m-
N8Gh2deSOZEGhxM3u5hV1Cgtk5LwPkGgzAWFpht_gsFu68wi05GAtDa5x8VWH_zoN-y4mn00-
cdhh2DV46QqBKKmJzG6eR7FK8YAaOlhJHiqVETwqDbIbRqWRPc_wMzi9WW44RF4WIiYNf2PgNLEG
U3L839PD0t5cUh0SdxBvr0IdAbQuUcMQR_5v-
bLWrrtd5PFyP1lcKejOABIRHH0lyB05O91XiqDhXzOkL7ADd4aIVjZYyXZRFoL7u5iR0B4cWslOprFSJuE
2JQljA0VhlMaM5fy3BgHOYVDkXMfdDu7NVN_GqIZTfQH7q-aBIpn0D7LRjcLG-jUQ6OL"
}
```

Il valore di "access_token" è il token di autorizzazione che il client dovrà passare alle successive chiamate.

2.2 Successive chiamate OAUTH

Il token deve essere passato ad ogni successiva chiamata come header HTTP Bearer Token.

```
HEADER Authorization: Bearer [token]
```

3 API

3.1 Facilities

Il metodo consente di ottenere l'elenco dei Parcheggi serviti dal sistema.

```
GET https://www.onecenter.info/api/DAZ/GetFacilities
```

Il metodo ritorna un elenco di Facility :

```
{
  "Facilities": [
    {
```

```
"IdCompany": 17,  
"FacilityID": 402057,  
"Description": "Moncalieri",  
"City": "Moncalieri",  
"Address": "Via Alfieri, 17/a",  
"ZIPCode": "10024",  
"Telephone1": "0116406583",  
"Telephone2": null
```

3.2 Facility Free Places

Il metodo ritorna lo stato di occupazione di tutte le aree di un Parcheggio, identificato dal parametro **FacilityID** (indicato per ogni parcheggio nella struttura ritornata dal metodo *GetFacilities*).

Per esempio:

```
GET https://www.onecenter.info/api/DAZ/FacilityFreePlaces?FacilityID=406884
```

Il metodo ritorna la struttura:

```
{  
  "FreePlaces": [  
    {  
      "FacilityID": 406884,  
      "ParkNo": 0,  
      "FacilityDescription": "Park P.zza Meda",  
      "CountingCategoryNo": 1,  
      "CountingCategory": "SostaBreve",  
      "FreeLimit": 467,  
      "OccupancyLimit": 468,  
      "CurrentLevel": 88,  
      "Reservation": 0,  
      "Capacity": 468  
    },  
    {  
      "FacilityID": 406884,  
      "ParkNo": 0,  
      "FacilityDescription": "Park P.zza Meda",  
      "CountingCategoryNo": 2,  
      "CountingCategory": "Abbonati",  
    }  
  ]  
}
```

```

    "FreeLimit": 9999,
    "OccupancyLimit": 9999,
    "CurrentLevel": 22,
    "Reservation": 0,
    "Capacity": 9999
  },
  [...]
],
"Status": "OK",
"Message": null
}

```

Il numero corrente di posti liberi è dato dal conteggio di:

Capacity - CurrentLevel

NB: i valori di Capacity a 9999 indicano una errata configurazione sulla DAZ.

E' possibile filtrare i posti liberi ritornati tramite il campo opzionale **CountingCategoryNo**.
Per esempio:

GET <https://www.onecenter.info/api/DAZ/FacilityFreePlaces?FacilityID=406884?CountingCategoryNo=3>

Il metodo ritorna la struttura:

```

{
  "FreePlaces": [
    {
      "FacilityID": 406884,
      "ParkNo": 0,
      "FacilityDescription": "Park P.zza Meda",
      "CountingCategoryNo": 3,
      "CountingCategory": "Totale",
      "FreeLimit": 467,
      "OccupancyLimit": 468,
      "CurrentLevel": 88,
      "Reservation": 0,
      "Capacity": 468
    },
  ],
  "Status": "OK",
  "Message": null
}

```



4 Utente: XX

```
client_id: XXX  
client_secret: xxx  
username: xxx@xpk.it  
password: XXXXXXXXXX
```