



QOE RATE PLAN MANAGEMENT USING REST API & PYTHON

Cambium Networks QoE

Version (1.3)



Contents

Introduction	3
Functionality	3
Python Script Configuration.....	4
QoE REST API Access Configuration	5
Tool Python Requirements	6
Script Usage	7
Rate Policy Management	8
Add or Update Rate Policy	8
Delete Rate Policy	8
Retrieve Rate Policy	9
Assign (Update) Subscriber's Rate Policy, Subscriber Groups and Quota	10
Retrieve Subscriber's Rate Policy.....	11
Delete Subscriber's Rate Policy.....	12
Retrieve Subscriber Metrics.....	12
Batch subscribers rate plan configuration from .csv file	16

Introduction

This tool allows the QoE user to manage subscriber rate plans from any system using REST APIs. QoE REST API documentation is available at Cambium Networks support site

(https://support.cambiumnetworks.com/files/qoe_qoe/).

The script is developed by Cambium Networks to illustrate how to use REST APIs to manage subscribers rate plans. The script can be updated by end users to meet their system configuration and network model.

Functionality

The tool provides the following functionalities:

- 1- Add Rate Policy (Service Plan)
- 2- Update Rate Policy (Service Plan)
- 3- Delete Rate Policy (Service Plan)
- 4- Retrieve Rate Policy (Service Plan)
- 5- Assign Subscriber to Rate Policy, Subscriber Groups and define quota.
- 6- Update Subscriber's Rate Policy
- 7- Delete Subscriber's Rate Policy
- 8- Retrieve Subscriber's Rate Policy
- 9- Retrieve Subscriber's Metrics
- 10- Batch subscribers rate plan configuration from .csv file

Python Script Configuration

There are 4 configuration parameters:

- The QoE_MNG_IP is the management IP address of the QoE server.
- The QoE_REST_PORT is the REST API Port number configure don the QoE.
- The QoE_REST_USER is the REST API Users username configure don the QoE.
- The QoE_REST_PASSWORD is the REST API Users password configure don the QoE.

There are 2 ways to configure those parameters:

- 1- Update the script: This is recommended if there is only one QoE server to be configured. The parameters can be changed in the script at the top section with the “To Do” title, the default values are:
 - QoE_MNG_IP = "10.0.0.100"
 - QoE_REST_PORT = "3443"
 - QoE_REST_USER = "qoe-rest-user"
 - QoE_REST_PASSWORD = "qoe-rest-passwd"
- 2- Through configuration file: The parameters can be saved in a configuration file and the configuration file name is passed as input parameter to the script. Each line of the file holds one parameter, and the parameters are in the following order:
 - Line #1: The QoE_MNG_IP
 - Line #2: The QoE_REST_PORT
 - Line #3: The QoE_REST_USER
 - Line #4: The QoE_REST_PASSWORD

The `-cfg` option is used to select the configuration file.

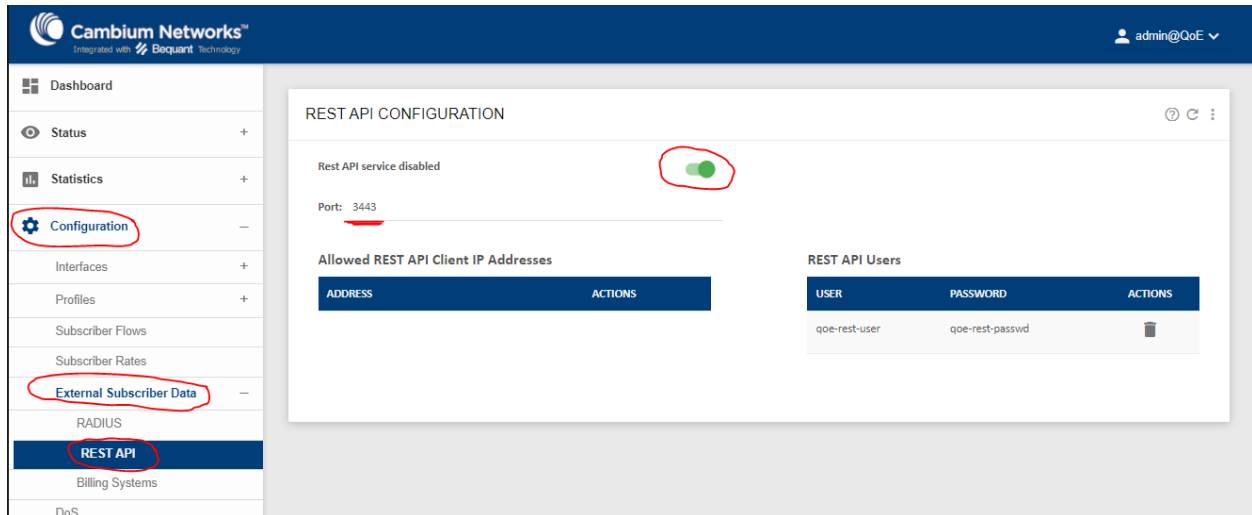
This option can be used if there are multiple QoE servers that the customer has. The customer will create multiple configuration files, one per QoE server. Then pass the appropriate config file to the script using the `-cfg` option.


Below is an example of the configuration file contents:

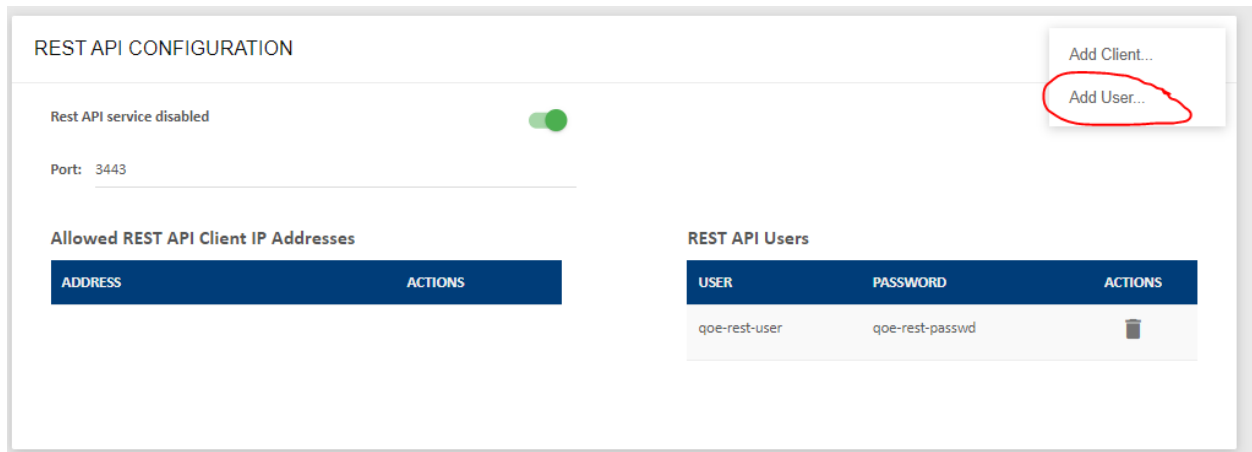
```
10.0.0.100
3443
qoe-rest-user
qoe-rest-passwd
```

QoE REST API Access Configuration

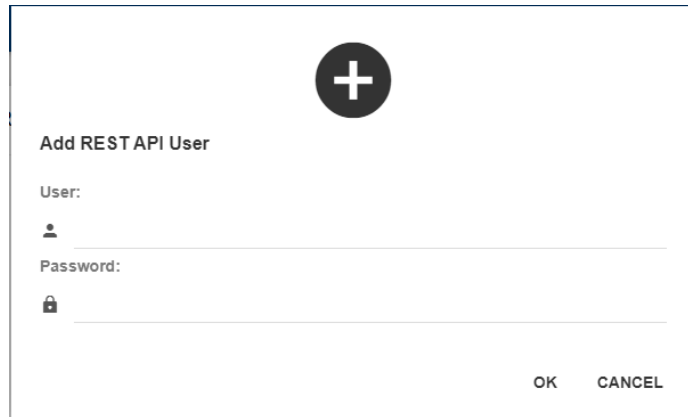
The following image shows an example of how-to setup a REST API User on the QoE



To add a REST API User click on the  and select Add User



The following windows will pop up:

A dialog box titled "Add REST API User" with a dark blue header bar containing a white plus icon. The dialog has two input fields: "User:" with a person icon and "Password:" with a lock icon. At the bottom right are "OK" and "CANCEL" buttons.

+

Add REST API User

User:

Password:

OK CANCEL

Enter the user and password and click on OK. Update the python script QoE_REST_USER and QoE_REST_PASSWORD with the configured user and password.

Tool Python Requirements

Python version 3.10 or above

Install argpars and optparse modules. (pip install argpars .. pip install optparse)

Script Usage

The script has a built in help option to display usage:

C:\python.exe QoE-REST-RatePlansManagement.py -h

usage: QoE-REST-RatePlansManagement.py [-h] [-p POLICYNAME] [-pi POLICYID] [-dl DOWNLINKRATE] [-ul UPLINKRATE] [-acm ACM] [-s SUBSCRIBER] [-si SUBSCRIBERID] [-m METRIC] [-mi METRIC_INTERVAL] [-mp METRIC_PERIOD] [-f SUBS_RATE_PLANS_FILE] [-cfg QOE_ACCESS_CONFIG_FILE] **action**

mandatory arguments:

action Action to be performed:

 addPolicy | getPolicy | deletePolicy | setSubRatePolicy | getSubRatePolicy | deleteSubRatePolicy | getSubMetrics | loadSubsFromFile

options:

-h, --help show this help message and exit

-p POLICYNAME, --policyName POLICYNAME

 Policy Name, no spaces or special characters

-pi POLICYID, --policyId POLICYID

 Policy Identifier, no spaces or special characters

-dl DOWNLINKRATE, --downlinkRate DOWNLINKRATE

 Downlink rate in kbps

-ul UPLINKRATE, --uplinkRate UPLINKRATE

 Uplink rate in kbps

-acm ACM, --acm ACM Automatic Congestion Management (ACM) enabled: true/false

-s SUBSCRIBER, --subscriber SUBSCRIBER

 subscriber IP address, IPv4 only

-si SUBSCRIBERID, --subscriberId SUBSCRIBERID

 subscriber Id

-m METRIC, --metric METRIC

 subscriber metric to retrieve: bandwidth|flows|latency|retransmission|volume

-mi METRIC_INTERVAL, --metric_interval METRIC_INTERVAL

 subscriber metric time interval in minutes (default 60 minutes)

-mp METRIC_PERIOD, --metric_period METRIC_PERIOD

 subscriber metric period in hours (default: 24 hours). The maximum query period is 3 months

-f SUBS_RATE_PLANS_FILE, --subs_rate_plans_file SUBS_RATE_PLANS_FILE

 subscriber rate plans file name

-cfg QOE_ACCESS_CONFIG_FILE, --qoe_access_config_file QOE_ACCESS_CONFIG_FILE

QoE REST Configuration file name

-grps SUB_GROUP_LIST, --sub_group_list SUB_GROUP_LIST

Subscriber group list up to 8 groups ':' separated.

-qt SUB_QUOTA_LIST, --sub_quota_list SUB_QUOTA_LIST

Subscriber quota info ':' separated, formatted as

ExpirationDate(YYYY-MM-DD):ExpirationTime(HH-MM):Volume(kBytes):VolumeIncrement(kBytes)

Rate Policy Management

Add or Update Rate Policy

Usage:

python.exe QoE-REST-RatePlansManagement.py addPolicy --policyName [polycynname] --policyId [policyID] --downlinkRate [dl_rate_kbps] --uplinkRate [ul_rate_kbps] --acm [true/false]

The following example adds (or updates) a rate policy with:

- Name: 100MDL_20MUL
- Policy ID: 100MDL_Policy
- DL Rate: 100 Mbps
- UL Rate: 20 Mbps
- ACM (Automatic Congestion Management feature): true (enabled)

C:\python.exe QoE-REST-RatePlansManagement.py addPolicy -p 100MDL_20MUL -pi 100MDL_Policy -dl 100000 -ul 20000 -acm true

Delete Rate Policy

Usage:

python.exe QoE-REST-RatePlansManagement.py deletePolicy --policyName [polycynname]

The following example deletes a rate policy named [100MDL_20MUL](#)


```
C:\python.exe QoE-REST-RatePlansManagement.py deletePolicy -p 100MDL_20MUL
```

The following example deletes ALL rate policies with Policy ID [100MDL_Policy](#)

```
C:\python.exe QoE-REST-RatePlansManagement.py deletePolicy -pi 100MDL_Policy
```

Retrieve Rate Policy

Usage:

```
python.exe QoE-REST-RatePlansManagement.py getPolicy [--policyName [policyname]]
```

The following example retrieves the rate policy named [100MDL_20MUL](#)

```
C:\python.exe QoE-REST-RatePlansManagement.py getPolicy -p 100MDL_20MUL
```

The output is in JSON format:

```
{
  "policyName": "100MDL_20MUL",
  "policyId": "100MDL_Policy",
  "rateLimitDownlink": {
    "rate": 100000,
    "congestionMgmt": true
  },
  "rateLimitUplink": {
    "rate": 20000
  }
}
```

The following example retrieves ALL rate policies

```
C:\python.exe QoE-REST-RatePlansManagement.py getPolicy
```

The output is in JSON format:

```
{
  "items": [
    {
      "policyName": "100MDL\_20MUL",
      "policyId": "100MDL_Policy",
      "rateLimitDownlink": {
        "rate": 100000,
        "congestionMgmt": true
      },
      "rateLimitUplink": {
        "rate": 20000
      }
    }
  ]
}
```

```

    },
    "policyName": "100MDL_30MUL",
    "policyId": "100MDL_Policy",
    "rateLimitDownlink": {
        "rate": 100000,
        "congestionMgmt": true
    },
    "rateLimitUplink": {
        "rate": 30000
    }
},
{
    "policyName": "100MDL_50MUL",
    "policyId": "100MDL_Policy",
    "rateLimitDownlink": {
        "rate": 100000,
        "congestionMgmt": true
    },
    "rateLimitUplink": {
        "rate": 50000
    }
}
]
}

```

Assign (Update) Subscriber's Rate Policy, Subscriber Groups and Quota

Usage:

`C:\python.exe QoE-REST-RatePlansManagement.py setSubRatePolicy --subscriber [IPv4] --subscriberId [ID] --policyName [policyname] -qt [Quota Details] -grps [Subscriber Groups]`

`-qt` [':' separated numbers formatted as

`ExpirationDate:ExpirationTime:Volume(kBytes):VolumeIncrement(kBytes)`. e.g. `2023-12-30:23-30:90000000:0` for 90GB quota expiring on Dec 30th 2023 at 11:30 PM]

`ExpirationDate` format: `YYYY-MM-DD` or `0` for no expiration date

`ExpirationTime` format: `HH-MM` or `0` for no expiration time. Set to `0` if no expiration date.

`-grps` [':' separated groups names up to 8 groups ':' separated e.g., `AP1NorthWest:Site1Noth: Tower1 North West`]

use `"` for group name that includes spaces.

`:` cannot be part of a group name'.

Optional parameters: `--subscriberId`, `-qt` and `-grps`

The following example assigns subscriber 10.0.0.133 to rate policy named [100MDL_20MUL](#)

```
C:\python.exe QoE-REST-RatePlansManagement.py setSubRatePolicy -s 10.0.0.133 -si John-Brown -p 100MDL_20MUL
```

The following example changes subscriber 10.0.0.133 rate policy to [100MDL_50MUL](#)

```
C:\python.exe QoE-REST-RatePlansManagement.py setSubRatePolicy -s 10.0.0.133 -si John-Brown -p 100MDL_50MUL
```

The following example sets subscriber 10.100.23.34 rate policy to [100MDL_40MUL](#) and assigns the subscriber to **Subscriber Groups**: AP1, Tower1, Site1 and City1, it also defines a 123.457 GB volume quota that expires on Dec 31st 2023 11:00 PM.

```
C:\python.exe QoE-REST-RatePlansManagement.py setSubRatePolicy -s 10.100.23.34 -si John-Merchant-2 -p 100MDL_40MUL -grps AP1:Tower1:Site1:City1 -qt 2028-12-31:23-00:123456789:0
```

Same example as above except the Subscriber Group name has spaces, so ' ' is used in the name.

```
C:\python.exe QoE-REST-RatePlansManagement.py setSubRatePolicy -s 10.100.23.34 -si John-Merchant-2 -p 100MDL_40MUL -grps AP1:Tower1:Site1:'City1 North Side' -qt 2028-12-31:23-00:123456789:0
```

Same example as above except no quota.

```
C:\python.exe QoE-REST-RatePlansManagement.py setSubRatePolicy -s 10.100.23.34 -si John-Merchant-2 -p 100MDL_40MUL -grps AP1:Tower1:Site1:'City1 North Side'
```

Same example as above with quota date only (time is optional and will default to 12:00 AM).

```
C:\python.exe QoE-REST-RatePlansManagement.py setSubRatePolicy -s 10.100.23.34 -si John-Merchant-2 -p 100MDL_40MUL -grps AP1:Tower1:Site1:'City1 North Side' -qt 2028-12-31::123456789:0
```

Retrieve Subscriber's Rate Policy

Usage:

```
python.exe QoE-REST-RatePlansManagement.py getSubRatePolicy --subscriber [IPv4]
```

The following example retrieves subscriber 10.0.0.133 rate policy

```
C:\python.exe QoE-REST-RatePlansManagement.py getSubRatePolicy -s 10.0.0.133
```

The output is in JSON format:

```
{
  "subscriberIp": "10.0.0.133",
  "subscriberId": "John-Brown",
```

```
"policyRate": "100MDL_50MUL"
}
```

The following example retrieves the rate policy of ALL subscribers configured through REST:

C:\python.exe QoE-REST-RatePlansManagement.py getSubRatePolicy

The output is in JSON format:

```
{
  "items": [
    {
      "subscriberIp": "10.0.0.133",
      "subscriberId": "John-Brown",
      "policyRate": "100MDL_50MUL"
    },
    {
      "subscriberIp": "10.0.0.100",
      "subscriberId": "Mike-Brown",
      "policyRate": "100MDL_20MUL"
    },
    {
      "subscriberIp": "10.0.0.90",
      "subscriberId": "Al-Harris",
      "policyRate": "100MDL_20MUL"
    }
  ]
}
```

Delete Subscriber's Rate Policy

Usage:

python.exe QoE-REST-RatePlansManagement.py deleteSubRatePolicy --subscriber [IPv4] | --subscriberId [ID]

The following example deletes the rate policy of subscriber 10.0.0.100 using the IP address

C:\python.exe QoE-REST-RatePlansManagement.py deleteSubRatePolicy -s 10.0.0.100

The following example deletes the rate policy of subscriber 10.0.0.90 using the subscriber ID

C:\python.exe QoE-REST-RatePlansManagement.py deleteSubRatePolicy -si Al-Harris

Retrieve Subscriber Metrics

Usage:

```
python.exe QoE-REST-RatePlansManagement.py getSubMetrics --subscriber [IPv4] --metric  
[bandwidth|flows|latency|retransmission|volume] --metric_interval [interval-minutes] --metric_period  
[period-hours]
```

The following example retrieves the volume metrics for subscriber 10.0.0.133 in the last 2 hours with interval 5 minutes:

```
C:\python.exe QoE-REST-RatePlansManagement.py getSubMetrics -s 10.0.0.133 -m volume -mp 2 -mi 5
```

The output is in JSON format:

```
{
  "subscriberIp": "10.0.0.133",
  "timestamp": [
    1663878000,
    1663878300,
    1663878600,
    1663878900,
    1663879200,
    1663879500,
    1663879800,
    1663880100,
    1663880400,
    1663880700,
    1663881000,
    1663881300,
    1663881600,
    1663881900,
    1663882200,
    1663882500,
    1663882800,
    1663883100,
    1663883400,
    1663883700,
    1663884000,
    1663884300,
    1663884600,
    1663884900,
    1663885200
  ],
  "dataDownlink": [
    0.9675,
    1.5055,
    1.399,
    1.4171,
    2.832,
```

```
1.0049,  
5.0151,  
1.1529,  
6.4514,  
65.889,  
0.6082,  
0.9383,  
0.9498,  
9.7952,  
1.2841,  
0.6781,  
124.3488,  
1.27,  
0.9199,  
2.028,  
22.6179,  
0.7469,  
4.8027,  
2.9811,  
1.1995  
,  
"dataUplink": [  
0.6642,  
0.8723,  
1.0162,  
1.3307,  
1.155,  
1.024,  
1.1205,  
1.9935,  
1.6216,  
1.6804,  
0.6573,  
0.8235,  
1.2274,  
1.3263,  
0.8746,  
0.753,  
2.0302,  
0.7017,  
0.9183,  
2.4477,  
1.24,  
0.8684,  
1.3673,  
1.1814,  
0.8274  
]
```

```
}
```

The entries show DL Volume and UL Volume in MB for each timestamp entry.

Batch subscribers rate plan configuration from .csv file

The script has a built in help option to display usage:

```
C:\python.exe QoE-REST-RatePlansManagement.py loadSubsFromFile -f subs_rate_plans_csv_file [-acm [true|false]] [-cfg QoE_Configuration_File]
```

By default, ACM is enabled; if the switch `-acm false` is passed to the script, then ACM will be disabled for all rate plans.

The `subs_rate_plans_csv_file` is formatted as a sequence of comma separated parameters that identify the rate plan details of a subscriber:

Cust_Number, Cust_Name, IPv4_Address, DL_Spd_Kbps, UL_Spd_Kbps, Quota, Subscriber-Groups

The first line is a header field which includes the names of the parameters.

The parameters (Cust_Number, Cust_Name) can be repeated in multiple lines in the file to support multiple packages (services) under one account. If the parameters (Cust_Number, Cust_Name) are repeated then the IPv4_Address parameter must be different for all entries with the same parameters (Cust_Number, Cust_Name)

The following table defines each field:

Parameter	Description
Cust_Number	Unique Customer Number. Spaces in this parameter will be replaced with “_”.
Cust_Name	Customer Name. Spaces in this parameter will be replaced with “_”.
IPv4_Address	IPv4 Address
DL_Spd_Kbps	Downlink Rate in Kbps
UL_Spd_Kbps	Uplink Rate in Kbps
Quota	Filed to define subscriber usage quota. ' :' separated numbers formatted as ExpirationDate:ExpirationTime:Volume(kBytes) :VolumeIncrement(kBytes) . e.g. 2023-12-30:23-30:90000000:0 for 90GB quota expiring on Dec 30th 2023 at 11:30 PM] ExpirationDate format: YYYY-MM-DD or 0 for no expiration date. ExpirationTime format: HH-MM or 0 for no expiration time. Set to 0 if no expiration date
Subscriber_groups	The names of the subscriber groups this subscriber belongs to. A maximum of 8 groups is allowed. ' :' separated groups names up to 8 groups ' :' separated e.g. AP1NorthWest:Site1Noth:'Tower1 North West'] use ‘ ’ for group name that includes spaces. ' :' cannot be part of a group name'.

Below is a sample rate plans .csv file content:

```
Cust_Number, Cust_Name, IPv4_Address, DL_Spd_Kbps, UL_Spd_Kbps, Quota, Subscriber_groups
2345,customer name 1,10.100.48.6,50000,10000,2023-12-20:0:60000000:0,AP1:Tower1:City1:Center1
2346,customer name 2,10.100.48.7,10000,2000,,AP1:Tower1:City1:Center1
2347,customer name 3,10.100.48.8,20000,4000,2023-12-25:0:60000000:0,AP1:Tower1:City1:Center1
2348,customer name 4,10.100.48.9,120000,4000,,AP2:Tower2:City1:Center1
2349,customer name 5,10.100.48.10,220000,4000,2023-12-17:0:60000000:0,AP2:Tower2:City1:Center1
2350,customer name 6,10.100.48.11,320000,4000,2023-12-13:0:60000000:0,AP3:Tower3:City2:Center2
2351,customer name 7,10.100.48.12,50000,10000,2023-12-20:0:60000000:0,
2352,customer name 8,10.100.48.13,150000,10000,2023-12-20:0:60000000:0,AP3:Tower3:City2:Center2
2353,customer name 9,10.100.48.14,250000,10000,2023-12-20:0:60000000:0,AP4:Tower4:City3:Center3
2354,customer name 10,10.100.48.15,350000,10000,,AP4:Tower4:City3:Center3
2355,customer name 11,10.100.48.16,250000,10000,,
2356,customer name 12,10.100.48.17,350000,10000,2023-12-20:0:60000000:0,AP4:Tower4:City3:Center3
2357,customer name 13,10.100.48.18,250000,10000,2023-12-20:0:60000000:0,AP3:Tower3:City2:Center2
2358,customer name 14,10.100.48.19,350000,10000,2023-12-20:0:60000000:0,AP3:Tower3:City2:Center2
2359,customer name 15,10.100.48.20,250000,10000,2023-12-20:0:60000000:0,AP2:Tower2:City1:Center1
2360,customer name 16,10.100.48.21,350000,10000,2023-12-20:0:60000000:0,AP2:Tower2:City1:Center1
2361,customer name 17,10.100.48.22,250000,10000,2023-12-20:0:60000000:0,AP4:Tower4:City3:Center3
2362,customer name 18,10.100.48.22,350000,10000,0:0:0:0,AP4:Tower4:City3:Center3
```