

# Use the Metered License Server public REST API to retrieve the usage records for your supervisor

## Purpose of this REST API endpoint

The data reported by your site's Metered License Agent is available to you should you wish to download this data in order to generate your own charts or reports in-house.

## Basic Usage

1. Do an initial retrieval of all data for a single supervisor, from the very first record ever reported to the most current record available.
2. Make a note of the 'id' from the last record in that dataset; you will use that value + 1 as the "starting id" the next time you ask for more records.
3. At least 15 minutes after you perform the initial retrieval of all records, do another retrieval with the "starting id" set to the starting id from the last step.

## Python Dependencies

You can write your script in any language you like, but my example is in Python. Should you choose to also use Python, you'll need the Python [Requests](#) module and a very current version of Python 2.7 to satisfy the TLS/SSL requirements. I work with Python 2.7.13, but you can use most versions of Python that ship with CentOS 7 or other modern linux distribution. I suggest using [Pip](#) to install Requests.

## Authentication and Authorization

You must supply credentials for a user account authorized to access the data for a given supervisor; these are the user name and password used to login to the Metered Portal, and can be the same credentials that your Metered Licence Agent uses to connect to <https://mls.pipelinefx.com>.

```
def login(username, password):
    """
    login and get a token which can be used for the next 6 hours
    """
    login_url = 'https://mls.pipelinefx.com/api/v2/login'

    response = requests.post(
        login_url,
        json={
            'username': username,
            'password': password
        }
    )
    token = response.json()['data']['token']
    assert token is not None
    return token
```

## Usage Record Retrieval

```

def get_records(token, macaddress, starting_id=0):
    url =
'https://mls.pipelinefx.com/api/v2/records?macaddress={}&starting_id={}'.format(macadd
ress, starting_id)

    # -----
    # get up to a full week's records for each request
    page_size = 1440 * 7
    # -----

    page_num = 1
    url_page_n = url + '&page={}&per_page={}'.format(page_num, page_size)
    logging.debug(url_page_n)

    response = requests.get(
        url_page_n,
        headers={
            'Authorization': 'Bearer {}'.format(token)
        }
    )
    if response.status_code != 200:
        raise ValueError(response['Reason'])

    records = list()
    while len(response.json()):
        records.extend(response.json())

        logging.info('Received records up to {ts_date}
{ts_time}'.format(*records[-1]))
        page_num += 1
        url_page_n = url + '&page={}&per_page={}'.format(page_num, page_size)
        logging.debug(url_page_n)
        response = requests.get(
            url_page_n,
            headers={
                'Authorization': 'Bearer {}'.format(token)
            }
        )
        if response.status_code != 200:
            raise ValueError(response['Reason'])

    return records

```

## Sample MySQL table for record storage

I recommend storing all the retrieved records in a MySQL database for ease of charting, reporting, determining the starting id of the next data retrieval, etc. The following should serve as a starting place for your table.

**Note:** this table will not be normalized (does not adhere to one or more normal forms) but creating a macaddress table and using the id from the macaddress table in this table to do INNER JOIN's is beyond the scope of the page. Also note that the **id** column is not an auto-increment, but please be aware that it will still be unique, as that id value comes from another table in which it **is** an auto-increment value.

```
CREATE TABLE license_usage (  
  id int(11) NOT NULL  
  , macaddress CHAR(17) NOT NULL  
  , ts_time time NOT NULL  
  , ts_date date NOT NULL  
  , lic_used int(11) NOT NULL  
  , lic_prepaid int(11) NOT NULL  
  , lic_metered int(11) NOT NULL  
  , PRIMARY KEY (id)  
  , UNIQUE KEY macaddr_date_time (macaddress,ts_date,ts_time)  
);
```