

```
# (C) Copyright 2014, 2015 Hewlett Packard Enterprise Development Company LP # Copyright 2017 Fujitsu LIMITED # # Licensed under the Apache License, Version 2.0 (the "License"); # you may not use this file except in compliance with the License. # You may obtain a copy of the License at # # http://www.apache.org/licenses/LICENSE-2.0 # # Unless required by applicable law or agreed to in writing, software # distributed under the License is distributed on an "AS IS" BASIS, # WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or # implied. # See the License for the specific language governing permissions and # limitations under the License.
```

```
import logging from time import sleep
```

```
from six.moves.urllib import parse
```

```
LOG = logging.getLogger(name)
```

```
class MonascaManager(object):
```

```
    base_url = None
```

```
    def __init__(self, client):
        self.client = client
```

```
    def _parse_body(self, body):
        if type(body) is dict:
            self.next = None
            for link in body['links']:
                if link['rel'] == 'next':
                    self.next = link['href']
            return body['elements']
        else:
            return body
```

```
    def _list_rec(self, path, dim_key=None, **kwargs):
        """Get a list of metrics."""
        url_str = self.base_url + path
        if dim_key and dim_key in kwargs:
            dim_str = self.get_dimensions_url_string(kwargs[dim_key])
            kwargs[dim_key] = dim_str
```

```
        if kwargs:
            url_str += '?%s' % parse.urlencode(kwargs, True)
```

```
        body = self.client.list(
            path=url_str
        )
```

```
        return body
        #return self._parse_body(body)
```

```
    def thereis_next(self, links):
        for link in links:
            if link['rel'] == 'next':
                offset = link['href'].split('offset=')[1]
                LOG.error('Offset: {0}'.format(offset))
                return offset
        return None
```

```
    def _list(self, path, dim_key=None, **kwargs):
```

```
lista = []
```

```
    offset = kwargs.get('offset', 0)
    while True:
        kwargs['offset'] = offset
        body = self._list_rec(path, dim_key=None, **kwargs)
        LOG.error('Raw body: {0}'.format(body))
        offset = self.thereis_next(body['links'])
        if offset:
            lista = lista + self._parse_body(body)
            sleep(1)
        else:
            break
    #continuar = False
```

return lista

```
@staticmethod
def get_dimensions_url_string(dimensions):
    dim_list = list()
    for k, v in dimensions.items():
        # In case user specifies a dimension multiple times
        if isinstance(v, (list, tuple)):
            v = v[-1]
        if v:
            dim_str = k + ':' + v
        else:
            dim_str = k
        dim_list.append(dim_str)
    return ','.join(dim_list)
```

From:
<https://chaconpiza.com/> - **Martin Chacon Piza Wiki**

Permanent link:
<https://chaconpiza.com/start?rev=1543829559>

Last update: **2021/03/11 08:12**

