Untitled

June 17, 2021

[7]:

```
from pathlib import Path
import json
import os
from tinydb import TinyDB
current_dir = Path(os.getcwd()).absolute()
results_dir = current_dir.joinpath('results')
kv_data_dir = results_dir.joinpath('kvdb')
kv_data_dir.mkdir(parents=True, exist_ok=True)

def _load_json(json_path):
    with open(json_path) as f:
        return json.load(f)

class DocumentDB(object):
    def __init__(self, db_path):
        ## You can use the code from the previous exmaple if you would like
        people_json = kv_data_dir.joinpath('people.json')
        visited_json = kv_data_dir.joinpath('visited.json')
        sites_json = kv_data_dir.joinpath('sites.json')
        measurements_json = kv_data_dir.joinpath('measurements.json')
        self._db_path = Path(db_path)
        self._db = None
        ## TODO
        self._load_db()

    def _get_site(self, site_id):
        return self._site_lookup[str(site_id)]
    def _get_measurements(self, person_id):
        measurements = []
        for values in self._measurements_lookup.values():
            measurements.extend([value for value in values if␣
 ↪str(value['person_id']) == str(person_id)])
        return measurements
    def _get_visit(self, visit_id):
        visit = self._visit_lookup.get(str(visit_id))
        site_id= self.visit_lookup.get(str(site_id))
        site= self.sites_lookup.get(str(latitude,longitude))
        ## TODO: site_id = ...
        ## TODO: site = ...
        visit['site'] = site
        return visit
    def _load_db(self):
        self._db = TinyDB(self._db_path)
    persons = self._person_lookup.items()
    for person_id, record in persons:
            measurements = self._get_measurements(person_id)
            visit_ids = set([measurement['visit_id'] for measurement in␣
↪measurements])
            visits = []
    for visit_id in visit_ids:
            visit = self._get_visit(visit_id)
            visit['measurements'] = [measurement for measurement in␣
↪measurements if visit_id == measurement['visit_id']]
            visits.append(visit)
            record['visits'] = visits
            self._db.insert(record)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-7-1f2456c3b4cc> in <module>
     12         return json.load(f)
     13 
---> 14 class DocumentDB(object):
     15     def __init__(self, db_path):
     16         ## You can use the code from the previous exmaple if you would␣
 ↪like

<ipython-input-7-1f2456c3b4cc> in DocumentDB()
     42     def _load_db(self):
     43         self._db = TinyDB(self._db_path)
---> 44     persons = self._person_lookup.items()
     45     for person_id, record in persons:
     46             measurements = self._get_measurements(person_id)

NameError: name 'self' is not defined
```

[2]:

```
db_path = results_dir.joinpath('patient-info.json')
if db_path.exists():
    os.remove(db_path)

db = DocumentDB(db_path)
```

```
---------------------------------------------------------------------------
```

```
AttributeError                            Traceback (most recent call last)
<ipython-input-2-bb790709a559> in <module>
      3     os.remove(db_path)
      4 
----> 5 db = DocumentDB(db_path)

<ipython-input-1-bd8e0b3d0ba5> in __init__(self, db_path)
     22         self._db = None
     23         ## TODO
---> 24         self._load_db()
     25 
     26     def _get_site(self, site_id):

<ipython-input-1-bd8e0b3d0ba5> in _load_db(self)
     40     def _load_db(self):
     41         self._db = TinyDB(self._db_path)
---> 42         persons = self._person_lookup.items()
     43         for person_id, record in persons:
     44             measurements = self._get_measurements(person_id)

AttributeError: 'DocumentDB' object has no attribute '_person_lookup'
```

```
[ ]:
```