documentdb

September 11, 2022

[85]:

```
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
        self._site_lookup = _load_json(sites_json)
        self._measurements_lookup = _load_json(measurements_json)
        self._visit_lookup = _load_json(visited_json)
        self._person_lookup = _load_json(people_json)
        self._load_db()

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

    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():
            if str(values['person_id']) == str(person_id):
                measurements.extend([values])
        return measurements

    def _get_visit(self, visit_id):
        visits = []
        sites = []
        for values in self._visit_lookup.values():
            if str(visit_id) == values['visit_id']:
                site_id = values['site_id']
                site = self._site_lookup.get(str(site_id))
                sites.extend([site])
                visits.extend([values, sites])
        return visits

    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)
            if len(measurements) != 0:
                for measurement in measurements:
                    visit_ids = set([measurement['visit_id']])
                    visits = []
                    for visit_id in visit_ids:
                        visit = self._get_visit(visit_id)
                        visit.append(measurement)
                        visits.append(visit)
                record['visits'] = visits
            self._db.insert(record)
```

[86]:

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

db = DocumentDB(db_path)
```

[ ]: