kvdb

March 30, 2021

[1]:
```
import json
from pathlib import Path
import os

import pandas as pd
import s3fs
```

[2]:
```
def read_cluster_csv(file_path, endpoint_url='https://storage.budsc.
↪midwest-datascience.com'):
   s3 = s3fs.S3FileSystem(
       anon=True,
       client_kwargs={'endpoint_url': endpoint_url}
      )
   return pd.read_csv(s3.open(file_path, mode='rb'))
```

[3]:
```
df = read_cluster_csv('data/external/tidynomicon/site.csv')
df.head(10)
```

[3]:

	site_id	latitude	longitude
0	DR-1	-49.85	-128.57
1	DR-3	-47.15	-126.72
2	MSK-4	-48.87	-123.40

[4]:
```
df_person = read_cluster_csv('data/external/tidynomicon/person.csv')
df_person.head(10)
```

[4]:

	person_id	personal_name	family_name
0	dyer	William	Dyer
1	pb	Frank	Pabodie
2	lake	Anderson	Lake
3	roe	Valentina	Roerich
4	danforth	Frank	Danforth

[5]:
```
current_dir = Path(os.getcwd()).absolute()
print(current_dir)
```

```
/home/jovyan
```

[6]:
```
results_dir = current_dir.joinpath('results')
print(results_dir)
```

```
/home/jovyan/results
```

[7]:
```
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)

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')
```

[8]:
```
class KVDB(object):
    def __init__(self, db_path):
        self._db_path = Path(db_path)
        self._db = {}
        self._load_db()

    def _load_db(self):
        if self._db_path.exists():
            with open(self._db_path) as f:
                self._db = json.load(f)

    def get_value(self, key):
        return self._db.get(key)

    def set_value(self, key, value):
        self._db[key] = value

    def save(self):
        with open(self._db_path, 'w') as f:
            json.dump(self._db, f, indent=2)
```

[9]:
```
def create_sites_kvdb():
    db = KVDB(sites_json)
    df = read_cluster_csv('data/external/tidynomicon/site.csv')
    for site_id, group_df in df.groupby('site_id'):
        db.set_value(site_id, group_df.to_dict(orient='records')[0])
    db.save()
```

[10]:
```
create_sites_kvdb()
```

[11]:
```
def create_people_kvdb():
    db = KVDB(people_json)
```

```
    ## TODO: Implement code
    df = read_cluster_csv('data/external/tidynomicon/person.csv')
    for person_id, group_df in df.groupby('person_id'):
        db.set_value(person_id, group_df.to_dict(orient='records')[0])
    db.save()
```

[12]:
```
create_people_kvdb()
```

[13]:
```
def create_visits_kvdb():
    db = KVDB(visited_json)
    ## TODO: Implement code
    df = read_cluster_csv('data/external/tidynomicon/visited.csv')
    group_columns = ['visit_id','site_id']

    for group_columns,group_df in df.groupby(group_columns):
        visit_id=str(group_columns[0])
        key=str(group_columns)
        db.set_value(visit_id,group_df.to_dict(orient='records')[0])
    db.save()
```

[14]:
```
create_visits_kvdb()
```

[15]:
```
def create_measurements_kvdb():
    db = KVDB(measurements_json)
    ## TODO: Implement code
    df = read_cluster_csv('data/external/tidynomicon/measurements.csv')
    group_columns = ['visit_id','person_id','quantity']
    for group_columns,group_df in df.groupby(group_columns):
        key = str(group_columns)
        db.set_value(key,group_df.to_dict(orient='records'))
    db.save()
```

[16]:
```
create_measurements_kvdb()
```

[ ]:
```

```