

```

1 {
2   "cells": [
3     {
4       "cell_type": "code",
5       "execution_count": 1,
6       "outputs": [],
7       "source": [
8         "from pathlib import Path\n",
9         "import json\n",
10        "import os\n",
11        "\n",
12        "import pandas as pd\n",
13        "from tinydb import TinyDB\n",
14        "\n",
15        "current_dir = Path(os.getcwd()).absolute()\n",
16        "results_dir = current_dir.joinpath('results')\n",
17        "\n",
18        "kv_data_dir = results_dir.joinpath('kvdb')\n",
19        "kv_data_dir.mkdir(parents=True, exist_ok=True)"
20      ],
21      "metadata": {
22        "collapsed": false,
23        "pycharm": {
24          "name": "#%%\n"
25        }
26      },
27      {
28        "cell_type": "code",
29        "execution_count": 2,
30        "outputs": [
31          {
32            "data": {
33              "text/plain": "4"
34            },
35            "execution_count": 2,
36            "metadata": {},
37            "output_type": "execute_result"
38          }
39        ],
40        "source": [

```

```

41     "people_json = kv_data_dir.joinpath('people.json
    ')\n",
42     "visited_json = kv_data_dir.joinpath('visited.
    json')\n",
43     "sites_json = kv_data_dir.joinpath('sites.json')\n
    n",
44     "measurements_json = kv_data_dir.joinpath('
    measurements.json')\n",
45     "\n",
46     "\n",
47     "paths = [\n",
48     "    people_json,\n",
49     "    visited_json,\n",
50     "    sites_json,\n",
51     "    measurements_json\n",
52     "]\n",
53     "\n",
54     "contents = []\n",
55     "\n",
56     "for path in paths:\n",
57     "    with open(path, 'r') as f:\n",
58     "        contents.append(json.loads(f.read()))\n
    ",
59     "\n",
60     "# print(type(json.dumps(contents)))\n",
61     "\n",
62     "\n",
63     "\n",
64     "db_json = json.dumps(contents)\n",
65     "type(contents[0])\n",
66     "len(contents)"
67 ],
68 "metadata": {
69     "collapsed": false,
70     "pycharm": {
71         "name": "#%%\n"
72     }
73 }
74 },
75 {
76     "cell_type": "code",

```

```

77     "execution_count": 3,
78     "metadata": {
79         "pycharm": {
80             "name": "#%%\n"
81         }
82     },
83     "outputs": [],
84     "source": [
85         "class DocumentDB(object):\n",
86         "    def __init__(self, db_path):\n",
87         "        ## You can use the code from the
previous example if you would like\n",
88         "        people_json = kv_data_dir.joinpath('
people.json')\n",
89         "        visited_json = kv_data_dir.joinpath('
visited.json')\n",
90         "        sites_json = kv_data_dir.joinpath('
sites.json')\n",
91         "        measurements_json = kv_data_dir.
joinpath('measurements.json')\n",
92         "\n",
93         "        self._db_path = Path(db_path)\n",
94         "        self._db = None\n",
95         "        ## TODO: Implement code\n",
96         "\n",
97         "        paths = [\n",
98         "            people_json,\n",
99         "            visited_json,\n100         "            sites_json,\n101         "            measurements_json\n",
102         "        ]\n",
103         "\n",
104         "        contents = []\n",
105         "\n",
106         "        for path in paths:\n",
107         "            with open(path, 'r') as f:\n",
108         "                contents.append(json.loads(f.
read()))\n",
109         "\n",
110         "        self._db_json = contents\n",
111         "        self._load_db()\n",

```

```

112     "\n",
113     "\n",
114     "    def _load_db(self):\n",
115     "        self._db = TinyDB(self._db_path)\n",
116     "        ## TODO: Implement code\n",
117     "        # self._db.insert(self._db_json)\n",
118     "\n",
119     "\n",
120     "        for x in range(len(self._db_json)):\n",
121     "            self._db.insert(self._db_json[x])"
122     ]
123 },
124 {
125     "cell_type": "code",
126     "execution_count": null,
127     "outputs": [],
128     "source": [
129         "db_path = results_dir.joinpath('patient-info."
130         "json')\n",
131         "if db_path.exists():\n",
132         "    os.remove(db_path)\n",
133         "\n",
134         "db = DocumentDB(db_path)"
135     ],
136     "metadata": {
137         "collapsed": false,
138         "pycharm": {
139             "name": "#%%\n",
140             "is_executing": true
141         }
142     }
143 },
144 "metadata": {
145     "kernel_spec": {
146         "display_name": "Python 3",
147         "language": "python",
148         "name": "python3"
149     },
150     "language_info": {
151         "codemirror_mode": {

```

```
152     "name": "ipython",
153     "version": 3
154 },
155 "file_extension": ".py",
156 "mimetype": "text/x-python",
157 "name": "python",
158 "nbconvert_exporter": "python",
159 "pygments_lexer": "ipython3",
160 "version": "3.8.3"
161 }
162 },
163 "nbformat": 4,
164 "nbformat_minor": 4
165 }
```