Trouble converting dataframe to json

Asked 1 month ago Modified 1 month ago Viewed 52 times

1

I have the following dataframe:

```
                                                 2023-06-30      2022-06-30
2021-06-30      2020-06-30
Ordinary Shares Number                          7432000000.0    7464000000.0
7519000000.0    7571000000.0
Share Issued                                    7432000000.0    7464000000.0
7519000000.0    7571000000.0
Net Debt                                       12533000000.0   35850000000.0
43922000000.0   49751000000.0
Total Debt                                     59965000000.0   61270000000.0
67775000000.0   70998000000.0
Tangible Book Value                           128971000000.0   87720000000.0
84477000000.0   67915000000.0
...                                                      ...             ...
...             ...
Cash Cash Equivalents And Short Term Investments  111262000000.0  104757000000.0
130334000000.0  136527000000.0
Other Short Term Investments                    76558000000.0   90826000000.0
116110000000.0  122951000000.0
Cash And Cash Equivalents                       34704000000.0   13931000000.0
14224000000.0   13576000000.0
Cash Equivalents                                26226000000.0    5673000000.0
6952000000.0             NaN
Cash Financial                                   8478000000.0    8258000000.0
7272000000.0             NaN

[73 rows x 4 columns]
```

And I am trying to convert it to json in the following format:

```
{
    "2023-06-30": {
        "Ordinary Shares Number": "7432000000.0",
        ...
    },
    "2022-06-30": {
        "Ordinary Shares Number": "7464000000.0",
        ...
    },
    "2021-06-30": {
        "Ordinary Shares Number": "7519000000.0",
        ...
    },
    "2020-06-30": {
        "Ordinary Shares Number": "7571000000.0",
        ...
    }
}
```

However my attempts to convert it have ranged from bad to worse so I really have no idea what I'm doing.

My attempts are either giving my a very undesirable json format or a type error about the timestamps:

For example:

```
out = json.dumps({c: dict(zip(balance.index, balance[c])) for c in balance.columns},
indent=4)
print(out)
```

Results in:

```
Traceback (most recent call last):
  File "/usr/lib/python3.8/runpy.py", line 194, in _run_module_as_main
    return _run_code(code, main_globals, None,
  File "/usr/lib/python3.8/runpy.py", line 87, in _run_code
    exec(code, run_globals)
  File "/home/jesse_b/tools/stonk-db/stonkdb/__main__.py", line 39, in <module>
    main()
  File "/home/jesse_b/tools/stonk-db/stonkdb/__main__.py", line 32, in main
    out = json.dumps({c: dict(zip(balance.index, balance[c])) for c in
balance.columns}, indent=4)
  File "/usr/lib/python3.8/json/__init__.py", line 234, in dumps
    return cls(
  File "/usr/lib/python3.8/json/encoder.py", line 201, in encode
    chunks = list(chunks)
  File "/usr/lib/python3.8/json/encoder.py", line 431, in _iterencode
    yield from _iterencode_dict(o, _current_indent_level)
  File "/usr/lib/python3.8/json/encoder.py", line 376, in _iterencode_dict
    raise TypeError(f'keys must be str, int, float, bool or None, '
TypeError: keys must be str, int, float, bool or None, not Timestamp
```

python json dataframe

Share Edit Follow Flag

edited Dec 31, 2023 at 23:56

asked Dec 31, 2023 at 23:49

jesse_b
149 1 9

2 Answers

Sorted by: Highest score (default)

1

You can use dict-comprehension:

```
import json

# convert the columns to string before (if needed)
# df.columns = df.columns.astype(str)
```

```
out = json.dumps({c: dict(zip(df.index, df[c])) for c in df.columns}, indent=4)
print(out)
```

Prints:


```
{
    "2023-06-30": {
        "Ordinary Shares Number": 7432000000.0,
        "Share Issued": 7432000000.0,
        "Net Debt": 12533000000.0,
        "Total Debt": 59965000000.0,
        "Tangible Book Value": 128971000000.0,
        "Cash Cash Equivalents And Short Term Investments": 111262000000.0,
        "Other Short Term Investments": 76558000000.0,
        "Cash And Cash Equivalents": 34704000000.0,
        "Cash Equivalents": 26226000000.0,
        "Cash Financial": 8478000000.0
    },
    "2022-06-30": {
        "Ordinary Shares Number": 7464000000.0,
        "Share Issued": 7464000000.0,
        "Net Debt": 35850000000.0,
        "Total Debt": 61270000000.0,
        "Tangible Book Value": 87720000000.0,
        "Cash Cash Equivalents And Short Term Investments": 104757000000.0,
        "Other Short Term Investments": 90826000000.0,
        "Cash And Cash Equivalents": 13931000000.0,
        "Cash Equivalents": 5673000000.0,
        "Cash Financial": 8258000000.0
    },
    "2021-06-30": {
        "Ordinary Shares Number": 7519000000.0,
        "Share Issued": 7519000000.0,
        "Net Debt": 43922000000.0,
        "Total Debt": 67775000000.0,
        "Tangible Book Value": 84477000000.0,
        "Cash Cash Equivalents And Short Term Investments": 130334000000.0,
        "Other Short Term Investments": 116110000000.0,
        "Cash And Cash Equivalents": 14224000000.0,
        "Cash Equivalents": 6952000000.0,
        "Cash Financial": 7272000000.0
    },
    "2020-06-30": {
        "Ordinary Shares Number": 7571000000.0,
        "Share Issued": 7571000000.0,
        "Net Debt": 49751000000.0,
        "Total Debt": 70998000000.0,
        "Tangible Book Value": 67915000000.0,
        "Cash Cash Equivalents And Short Term Investments": 136527000000.0,
        "Other Short Term Investments": 122951000000.0,
        "Cash And Cash Equivalents": 13576000000.0,
        "Cash Equivalents": NaN,
        "Cash Financial": NaN
    }
}
```


edited Jan 1 at 0:04

answered Dec 31, 2023 at 23:54

Andrej Kesely
183k 15 49 92

Thanks, I hadn't tried that but it is giving me the same error that a few of my attempts did which is: `TypeError: keys must be str, int, float, bool or None, not Timestamp` – jesse_b Dec 31, 2023 at 23:55

@jesse_b You can convert the row with timestamp to string before converting to Json, e.g. `df.loc["Timestamp Row"] = df.loc["Timestamp Row", :].astype(str)` – Andrej Kesely Dec 31, 2023 at 23:57

apologies as I'm sure I'm missing something that is obvious to most but I'm pretty new to python. Getting this: `File "/home/jesse_b/.local/lib/python3.8/site-packages/pandas/core/indexes/base.py", line 3655, in get_loc raise KeyError(key) from err ; KeyError: 'Timestamp Row'` – jesse_b Dec 31, 2023 at 23:59

@jesse_b Or try to convert the columns to string: `df.columns = df.columns.astype(str)` – Andrej Kesely Jan 1 at 0:00

1 That did it! Thank you so much. You're a gentleman, a scholar, and a breeder of fine horses. – jesse_b Jan 1 at 0:01

1

We can achieve the same JSON structure by just setting the first column as the index of the DataFrame and then converting it to JSON using `df.to_json(orient='columns')`. In the `'columns'` orientation, the JSON takes the shape where each column in the DataFrame becomes a key, and the corresponding values are stored in an array under that key.

```
import pandas as pd
import json

df.set_index(df.columns[0], inplace=True)
result = df.to_json(orient='columns')

parsed = json.loads(result)
print(json.dumps(parsed, indent=4))
```

Prints


```
{
    "2023-06-30": {
        "Ordinary Shares Number": 7432000000.0,
        "Share Issued": 7432000000.0,
        "Net Debt": 12533000000.0,
        "Total Debt": 59965000000.0,
        "Tangible Book Value": 129000000000.0,
        "Cash Cash Equivalents": 111000000000.0,
        "Other Short Term Investme": 76558000000.0,
        "Cash And Cash Equivalents": 34704000000.0,
        "Cash Equivalents": 26226000000.0,
        "Cash Financial": 8478000000.0
    },
    "2022-06-30": {
```

```
            "Ordinary Shares Number": 7464000000.0,
            "Share Issued": 7464000000.0,
            "Net Debt": 35850000000.0,
            "Total Debt": 61270000000.0,
            "Tangible Book Value": 87720000000.0,
            "Cash Cash Equivalents": 105000000000.0,
            "Other Short Term Investme": 90826000000.0,
            "Cash And Cash Equivalents": 13931000000.0,
            "Cash Equivalents": 5673000000.0,
            "Cash Financial": 8258000000.0
        },
        "2021-06-30": {
            "Ordinary Shares Number": 7519000000.0,
            "Share Issued": 7519000000.0,
            "Net Debt": 43922000000.0,
            "Total Debt": 67775000000.0,
            "Tangible Book Value": 84477000000.0,
            "Cash Cash Equivalents": 130000000000.0,
            "Other Short Term Investme": 116000000000.0,
            "Cash And Cash Equivalents": 14224000000.0,
            "Cash Equivalents": 6952000000.0,
            "Cash Financial": 7272000000.0
        },
        "2020-06-30": {
            "Ordinary Shares Number": 7571000000.0,
            "Share Issued": 7571000000.0,
            "Net Debt": 49751000000.0,
            "Total Debt": 70998000000.0,
            "Tangible Book Value": 67915000000.0,
            "Cash Cash Equivalents": 137000000000.0,
            "Other Short Term Investme": 123000000000.0,
            "Cash And Cash Equivalents": 13576000000.0,
            "Cash Equivalents": null,
            "Cash Financial": null
        }
    }
```


Share Edit Delete Flag

edited Jan 1 at 7:39

answered Jan 1 at 0:57

Chaithra KC

31 4