How can a Pandas DataFrame be converted to JSON?

Authored by

stats writer

April 17, 2024

A Pandas DataFrame can be converted to JSON by using the "to_json()" function. This function takes in various parameters such as the desired file path, orientation, and data format to convert the DataFrame into a JSON file. By default, the function converts the entire DataFrame into a JSON object with column names as keys and row values as values. However, the function also allows for customization, such as selecting specific columns or rows to be converted, and choosing the desired orientation of the JSON output. The resulting JSON file can then be easily accessed and utilized for data analysis or sharing with other applications.

Convert a Pandas DataFrame to JSON

Often you might be interested in converting a pandas DataFrame to a JSON format.

Fortunately this is easy to do using the function, which allows you to convert a DataFrame to a JSON string with one of the following formats:

'split' : dict like {'index' -> , 'columns' -> , 'data' -> }'records' : list like 'index' : dict like {index -> {column -> value}}'columns' : dict like {column -> {index -> value}}'values' : just the values array'table' : dict like {'schema': {schema}, 'data': {data}}

This tutorial shows how to convert a DataFrame to each of the six formats using the following pandas DataFrame:

import pandas as pd

```
#create DataFrame
df = pd.DataFrame({'points': ,
'assists': })

#view DataFrame
df

points assists
0 25 5
1 12 7
2 15 7
3 19 12
```

Method 1: 'Split'

```
df.to_json(orient='split')

{
"columns": ,
"index": ,
"data": ,
,
,

]
}
```

Method 2: 'Records'

```
df.to_json(orient='records')
```

Method 3: 'Index'

```
df.to_json(orient='index')
```

```
{
"0": {
"points": 25,
"assists": 5
},
"1": {
"points": 12,
"assists": 7
},
"2": {
"points": 15,
"assists": 7
},
"3": {
"points": 19,
"assists": 12
}
```

}

Method 4: 'Columns'

df.to_json(orient='columns')

{
"points": {
"0": 25,
"1": 12,
"2": 15,
"3": 19
},
"assists": {
"0": 5,
"1": 7,
"2": 7,
"3": 12
}
}

Method 5: 'Values'

df.to_json(orient='values')

,

,

,

]

Method 6: 'Table'

df.to_json(orient='table')

{
"schema": {
"fields": ,
"primaryKey": ,
"pandas_version": "0.20.0"
},
"data":
}

How to Export a JSON File

You can use the following syntax to export a JSON file to a specific file path on your computer:

#create JSON file
json_file = df.to_json(orient='records')

```
#export JSON file
with open('my_data.json', 'w') as f:
f.write(json_file)
```

You can find the complete documentation for the pandas to_json() function .