

CSVデータをElasticsearchサーバーに移行するプログラムの開発

祖父江匠真

1 はじめに

今回は、(ここにリソース元をより詳しく明記する)DougoSyou 配下にある、CSV形式で保存された太陽光発電の環境データを Elasticsearch サーバーに移行するプログラムを開発した。

2 CSVデータをElasticsearchサーバーに移行するプログラムの開発

今回、データ移行に使用した CSV データの一部を図1に示す。図1の CSV データには、TIME、日射強度、外気温度、直流電力、交流電力のカラムが存在する。なお、TIME は計測日時であると推定される。

	標準	標準	標準	標準	標準
1	TIME	日射強度	外気温度	直流電力	交流電力
2	分	kw/ m ²	°C	kw	kw
3	2018/09/01 00:01	0.00	28.2	0.00	0.00
4	2018/09/01 00:02	0.00	28.2	0.00	0.00
5	2018/09/01 00:03	0.00	28.2	0.00	0.00
6	2018/09/01 00:04	0.00	28.2	0.00	0.00
7	2018/09/01 00:05	0.00	28.2	0.00	0.00
8	2018/09/01 00:06	0.00	28.2	0.00	0.00
9	2018/09/01 00:07	0.00	28.2	0.00	0.00
10	2018/09/01 00:08	0.00	28.2	0.00	0.00
11	2018/09/01 00:09	0.00	28.2	0.00	0.00

図1: 移行元の太陽光発電の環境データ

それぞれのカラムは、現在リサイクル館から送信されている太陽光発電の環境データを保存する Elasticsearch サーバーのインデックスにおける、JPtime, solarIrradiance(kw/m²), airTemperature(°C), dc-pw(kw), ac-pw(kw) フィールドと対応する。

ソースコード 1 は、移行先となる Elasticsearch サーバーのインデックスを作成するプログラムである。既存の Elasticsearch サーバーのインデックスとの互換性を考慮して、今回開発する移行プログラムでインデックスに追加するドキュメントのフィールド名は、JPtime, solarIrradiance(kw/m²), airTemperature(°C), dc-pw(kw), ac-pw(kw) としている。

ソースコード 1: CSV データの移行先となるインデックスを作成するプログラム

```
1 from elasticsearch import Elasticsearch
2
3 # Elasticsearch クライアント作成
4 es = Elasticsearch("http://localhost:9200")
5
6 # インデックス一覧の取得
7 indices = es.cat.indices(index="*", h="index").splitlines()
8 # 一度すべてのインデックスを削除する
9 for index in indices:
10     es.indices.delete(index=index)
11
12 # マッピングを作成
13 mapping = {
14     "mappings": {
15         "properties": {
16             "JPtime": {"type": "date"},
17             "solarIrradiance(kw/m^2)": {"type": "float"},
18             "airTemperature( )": {"type": "float"},
19             "dc-pw(kw)": {"type": "float"},
20             "ac-pw(kw)": {"type": "float"},
21         }
22     }
23 }
24 # マッピングを指定してインデックスを作成
25 es.indices.create(index="solars", body=mapping)
26
27 # 内部接続を閉じる
28 es.close()
```

ソースコード 2 は、ソースコード 1 で作成したインデックスに CSV データを移行するプログラムである。Python の datetime モジュールで生成される Date オブジェクトを、デフォルト設定のまま Elasticsearch に保存すると、Kibana 上で UTC 時間として扱われる。更に、ブラウザ環境からタイムゾーンを Asia/Tokyo と推定し、自動的に UTC 時間から JST 時間に変換するので 9 時間進んで表示されてしまう。この現象の対策として、ソースコード 2 の 26 行目では、Date オブジェクトを作成する際に、JST 時間に設定している。

ソースコード 2: CSV 形式の太陽光発電データを Elasticsearch サーバーに移行するプログラム

```
1 from elasticsearch import Elasticsearch
2 import pandas as pd
3 import codecs
4 import datetime
5
```

```

6 # Elasticsearch クライアント作成
7 es = Elasticsearch("http://localhost:9200")
8
9 filepath = "data/DougoSyou/1809010000.csv"
10 with codecs.open(filepath, "r", "Shift-JIS", "ignore") as file:
11     df = pd.read_csv(file, delimiter=",", skiprows=[1])
12
13     for i, row in df.iterrows():
14         year = int(row["TIME"][:4])
15         month = int(row["TIME"][5:7])
16         day = int(row["TIME"][8:10])
17         hour = int(row["TIME"][11:13])
18         minute = int(row["TIME"][14:])
19         row = {
20             "JPtime": datetime.datetime(
21                 year,
22                 month,
23                 day,
24                 hour,
25                 minute,
26                 tzinfo=datetime.timezone(datetime.timedelta(hours=9)),
27             ),
28             "solarIrradiance(kw/m^2)": float(row["日射強度"]),
29             "airTemperature( )": float(row["外気温度"]),
30             "dc-pw(kw)": float(row["直流電力"]),
31             "ac-pw(kw)": float(row["交流電力"]),
32         }
33         es.create(index="solars", id=i + 1, document=row)
34
35 # 内部接続を閉じる
36 es.close()

```

ソースコード 2 の実行によってインデックスに追加されたドキュメントを図 2 に示す。図 2 より、ドキュメント数が 43199 であることが分かるが、今回使用した CSV データは 2018 年 9 月 1 日から 2018 年 9 月 30 日までの間、1 分ごとに計測されたデータであるので、計測回数とドキュメント数が一致していることから、全ての CSV データを正しく Elasticsearch サーバーに移行できたことが分かる。

43,199 hits

Sort fields

	Document
<input checked="" type="checkbox"/>	ac-pw(kw) 0 airTemperature(°C) 25.3 dc-pw(kw) 0.19 Jptime Sep 1, 2018 @ 06:56:00.000 solarIrradiance(kw/m^2) 0.02 _id 416 _index solars _score 1
<input checked="" type="checkbox"/>	ac-pw(kw) 0.05 airTemperature(°C) 25.3 dc-pw(kw) 0.2 Jptime Sep 1, 2018 @ 06:57:00.000 solarIrradiance(kw/m^2) 0.02 _id 417 _index solars _score 1
<input checked="" type="checkbox"/>	ac-pw(kw) 0 airTemperature(°C) 25.3 dc-pw(kw) 0 Jptime Sep 1, 2018 @ 06:58:00.000 solarIrradiance(kw/m^2) 0.02 _id 418 _index solars _score 1
<input checked="" type="checkbox"/>	ac-pw(kw) 0 airTemperature(°C) 25.1 dc-pw(kw) 0 Jptime Sep 1, 2018 @ 06:59:00.000 solarIrradiance(kw/m^2) 0.02 _id 419 _index solars _score 1
<input checked="" type="checkbox"/>	ac-pw(kw) 0 airTemperature(°C) 25.2 dc-pw(kw) 0 Jptime Sep 1, 2018 @ 07:00:00.000 solarIrradiance(kw/m^2) 0.02 _id 420 _index solars _score 1
<input checked="" type="checkbox"/>	ac-pw(kw) 0 airTemperature(°C) 25.2 dc-pw(kw) 0 Jptime Sep 1, 2018 @ 07:01:00.000 solarIrradiance(kw/m^2) 0.02 _id 421 _index solars _score 1
<input checked="" type="checkbox"/>	ac-pw(kw) 0 airTemperature(°C) 25.1 dc-pw(kw) 0 Jptime Sep 1, 2018 @ 07:02:00.000 solarIrradiance(kw/m^2) 0.02 _id 422 _index solars _score 1
<input checked="" type="checkbox"/>	ac-pw(kw) 0.15 airTemperature(°C) 25 dc-pw(kw) 0.25 Jptime Sep 1, 2018 @ 07:03:00.000 solarIrradiance(kw/m^2) 0.03 _id 423 _index solars _score 1
<input checked="" type="checkbox"/>	ac-pw(kw) 0.21 airTemperature(°C) 25 dc-pw(kw) 0.31 Jptime Sep 1, 2018 @ 07:04:00.000 solarIrradiance(kw/m^2) 0.03 _id 424 _index solars _score 1

図 2: 移行後の太陽光発電の環境データ

3 おわりに

今回は, DougoSyou 配下にある, CSV 形式で保存された太陽光発電の環境データを Elasticsearch サーバーに移行するプログラムを開発した. 次回は, DougoSyou 配下にある全ての CSV データを Elasticsearch サーバーに移行するプログラムを開発する.

参考文献

- [1] さっと, ”Python Elasticsearch 基本的な使い方まとめ”,
https://qiita.com/satto_sann/items/8a63761bbfd6542bb9a2, 参照 May 8, 2022.