

Pandas, PyMongo

✓ Чтение и запись данных, часть 2

Практическая работа

Описание практической работы и формат сдачи

Выполните предложенные задачи: сохраните блокнот отдельно себе на диск и после этого пишите код в ячейках под задачами. Итоговый файл в формате `.ipynb` (файл Jupyter Notebook) загрузите в личный кабинет и отправьте на проверку.

```
[ ] import urllib.request as reg
import json

API_KEY = "a35c1eadcf824626961161619250306"
days = 2
url = f"http://api.weatherapi.com/v1/forecast.json?key={API_KEY}&q=London&days={days}"
session = reg.urlopen(url)
with reg.urlopen(url) as session:
    response = session.read().decode()
data = json.loads(response)
data

{'location': {'name': 'London',
              'region': 'City of London, Greater London',
              'country': 'United Kingdom',
              'lat': 51.5171,
              'lon': -0.1062,
              'tz_id': 'Europe/London',
              'localtime_epoch': 1749314611,
              'localtime': '2025-06-07 17:43'},
 'current': {'last_updated_epoch': 1749313800,
             'last_updated': '2025-06-07 17:30',
             'temp_c': 13.4,
             'temp_f': 56.1,
             'is_day': 1,
             'condition': {'text': 'Moderate rain',
                           'icon': '//cdn.weatherapi.com/weather/64x64/day/302.png',
                           'code': 1100}}
```

✓ Задача 2

Сохраните данные погоды, полученные в предыдущей задаче, в файл `weather.json`. При сохранении используйте четыре пробела для форматирования.

```
[ ] import json

[ ] with open("weather.json", "w") as file:
    json.dump(data, file, indent=4)
```

✓ Задача 3

Из содержимого файла `weather.json` получите **почасовой** прогноз погоды, загрузите его в `DataFrame` и выведите первые пять строк `DataFrame` на экран.

```
[ ] import pandas as pd

[ ] with open("weather.json", "r") as file:
    data = json.load(file)

[ ] hourly_forecast = []
for day in data['forecast']['forecastday']:
    hourly_forecast.extend(day['hour'])

[ ] df = pd.DataFrame(hourly_forecast)

[ ] df.head()
```

	time_epoch	time	temp_c	temp_f	is_day	condition	wind_mph	wind_kph	wind_degree	wind_dir	...	dewpoint_f	will_it_rain	chance_of_rain	will_it_snow	chance_of_snow	vis_km	vis_miles	gust_mph	gust_kph	uv
0	1749250800	2025-06-07 00:00	11.7	53.0	0	(text: 'Patchy rain nearby', 'icon': '//cdn...	7.6	12.2	213	SSW	...	48.7	1	100	0	0	10.0	6.0	11.7	18.8	0.0
1	1749254400	2025-06-07 01:00	11.6	53.0	0	(text: 'Patchy rain nearby', 'icon': '//cdn...	6.3	10.1	208	SSW	...	48.4	1	100	0	0	10.0	6.0	9.4	15.0	0.0
2	1749258000	2025-06-07 02:00	11.1	52.0	0	(text: 'Light drizzle', 'icon': '//cdn.weath...	4.9	7.9	211	SSW	...	49.4	1	100	0	0	2.0	1.0	7.2	11.5	0.0
3	1749261600	2025-06-07 03:00	11.1	52.0	0	(text: 'Light rain', 'icon': '//cdn.weathera...	3.8	6.1	217	SW	...	50.7	1	100	0	0	9.0	5.0	5.7	9.1	0.0
4	1749265200	2025-06-07 04:00	11.1	51.9	0	(text: 'Light rain', 'icon': '//cdn.weathera...	2.0	3.2	230	SW	...	50.7	1	100	0	0	9.0	5.0	3.0	4.8	0.0

5 rows × 24 columns

▼ Задача 4

Сохраните DataFrame из предыдущей задачи в файл `result.csv`. В качестве названия столбца-индекса используйте `index`.

```
[ ] df.to_csv('result.csv', index_label = 'index')
```

▼ Задача 5

Загрузите содержимое SQL-таблицы `ratings` в DataFrame. Выведите первые пять строк DataFrame на экран. Для подключения к БД используйте строку подключения, указанную в следующей ячейке.

```
[ ] import sqlalchemy
import psycopg2
import pandas as pd
from sqlalchemy import text
```

```
[ ] connection = 'postgresql+psycopg2://readonly:6hajV34RTQfmXh5@dsstudents.skillbox.ru:5433/db_ds_students'
engine = sqlalchemy.create_engine(connection)
connect = engine.connect()
df = pd.read_sql(text('select * from ratings'), connect)
df.head(5)
```

Exception during reset or similar
Traceback (most recent call last):
File "C:\Users\User\anaconda3\Lib\site-packages\sqlalchemy\pool\base.py", line 987, in _finalize_fairy
fairy._reset()
File "C:\Users\User\anaconda3\Lib\site-packages\sqlalchemy\pool\base.py", line 1433, in _reset
pool._dialect.do_rollback(self)
File "C:\Users\User\anaconda3\Lib\site-packages\sqlalchemy\engine\default.py", line 700, in do_rollback
dbapi_connection.rollback()
psycopg2.OperationalError: server closed the connection unexpectedly
This probably means the server terminated abnormally
before or while processing the request.

	userid	movieid	rating	timestamp
0	5953	83374	3.0	1432706275
1	5953	83480	3.0	1432706275
2	5953	83613	3.0	1432706272
3	5953	84152	3.0	1432706278
4	5953	84392	4.0	1432706275

▼ Задача 6

Сохраните первые 10000 записей датафрейма из предыдущей задачи в файл `result2.xlsx`, при сохранении задайте листу имя `MySheet`.

(Примечание: для указания имени листа в файле Excel как при чтении, так и при записи используется параметр `sheet_name`)

```
[ ] df = df.head(10000)
df.to_excel('result2.xlsx', sheet_name='MySheet')
```

✓ Задача 7

Определите количество документов в коллекции `users` в MongoDB. Если количество документов < 1000 , то в следующей ячейке загрузите всё содержимое этой коллекции в список и выведите нулевой элемент этого списка на экран. Для подключения к MongoDB используйте строку подключения, указанную в следующей ячейке.

```
[ ] !pip install pymongo==3.12.0
```

```
Collecting pymongo==3.12.0
  Downloading pymongo-3.12.0.tar.gz (818 kB)
    818.6/818.6 kB 12.8 MB/s eta 0:00:00
    Preparing metadata (setup.py) ... done
Building wheels for collected packages: pymongo
  Building wheel for pymongo (setup.py) ... done
  Created wheel for pymongo: filename=pymongo-3.12.0-cp311-cp311-linux_x86_64.whl size=510592 sha256=b4639ba3f1424135c740f634
  Stored in directory: /root/.cache/pip/wheels/0a/e0/8a/adfa1aa746c3a9984d16be698311d2ba894032a3ca7c400ed
Successfully built pymongo
Installing collected packages: pymongo
  Attempting uninstall: pymongo
    Found existing installation: pymongo 4.13.0
    Uninstalling pymongo-4.13.0:
      Successfully uninstalled pymongo-4.13.0
Successfully installed pymongo-3.12.0
```

```
[ ] import pymongo
```

```
[ ] conn = 'mongodb://students:X63673t47G103Sq@dsstudents.skillbox.ru:27017/?authSource=movies'
connect = pymongo.MongoClient(conn)
db = connect['movies']
db.list_collection_names() #какие коллекции доступны
```

```
['tags', 'users']
```

```
users = db['users']
head = users.find()
db.get_collection("users").estimated_document_count() #кол-во документов
```

```
102
```

```
[ ] users = db['users']
# head = users.find()
users.estimated_document_count() #кол-во документов
```

```
102
```

```
users = db['users']
head = users.find()
my_list = list(head)
my_list[0]
```

```
{'_id': ObjectId('5cc69095067496679c74f7a4'),
 'user_id': 'ed5da480-57a2-489c-8e68-512155ab9892',
 'first_name': 'Жанна',
 'last_name': 'Петрова',
 'mail': 'zhanna_03816@gmail.com',
 'age': 20.0}
```

Задача 8

Преобразуйте список всех документов коллекции `users` из предыдущей задачи в pandas DataFrame и сохраните его содержимое в файл `result3.xlsx`.

```
import pandas as pd
```

```
df = pd.DataFrame(my_list)
df.to_excel('result3.xlsx')
```