

1. Топіки

The screenshot shows a Python IDE with a file explorer on the left containing files like `goit-de-hw-05`, `configs.py`, `consumer.py`, `create_topics.py`, and `producer.py`. The main editor displays the `create_topics.py` file with the following code:

```
1 from kafka.admin import KafkaAdminClient, NewTopic
2 from configs import kafka_config
3
4 def create_topic(name, partitions, replication):
5     admin_client = KafkaAdminClient(
6         bootstrap_servers=kafka_config['bootstrap_servers'],
7         security_protocol=kafka_config['security_protocol'],
8         sasl_mechanism=kafka_config['sasl_mechanism'],
9         sasl_plain_username=kafka_config['username'],
10        sasl_plain_password=kafka_config['password']
11    )
12
13    topic = NewTopic(name=name, num_partitions=partitions, replication_factor=replication)
14    try:
15        admin_client.create_topics(new_topics=[topic], validate_only=False)
16        print(f"Topic '{name}' created successfully.")
17    except Exception as e:
18        print(f"An error occurred while creating topic {name}: {e}")
19    finally:
20        admin_client.close()
21
22 create_topic('Ivan_building_sensors', 3, 1)
23 create_topic('Ivan_temperature_alerts', 1, 1)
24 create_topic('Ivan_humidity_alerts', 1, 1)
25
```

The right sidebar shows the Variable Explorer with the `kafka_config` dictionary:

```
{'bootstrap_servers': ['localhost:9092'], 'username': '', 'password': '', ...}
```

The Console window at the bottom shows the output of the script:

```
Python 3.8.20 | packaged by conda-forge | (default, Sep 30 2024, 17:44:03) [MSC v.1929 64 bit (AMD64)]
Type "copyright", "credits" or "license()" for more information.
IPython 8.12.2 -- An enhanced Interactive Python. Type '?' for help.

In [1]: %runfile C:/Work/GoIT/DataEngineering/goit-de-hw-05/create_topics.py --wdir
Reloaded modules: numpy._utils._conversions, numpy._utils, numpy._globals, numpy._expired_attrs_2_0, numpy.version, numpy._distributor_init, numpy._utils._inspect
Topic 'Ivan_building_sensors' created successfully.
Topic 'Ivan_temperature_alerts' created successfully.
Topic 'Ivan_humidity_alerts' created successfully.

In [2]:
```

The screenshot shows a terminal window with the following command and output:

```
C:\Work\kafka_2.12-3.9.0\bin\windows>kafka-topics.bat --list --bootstrap-server localhost:9092
Ivan_building_sensors
Ivan_humidity_alerts
Ivan_temperature_alerts
building_sensors
humidity_alerts
temperature_alerts

C:\Work\kafka_2.12-3.9.0\bin\windows>
```

2. Генерація даних

The screenshot shows a Python IDE with a file explorer on the left containing files like `goit-de-hw-05`, `configs.py`, `consumer.py`, `create_topics.py`, and `producer.py`. The main editor displays the `producer.py` file with the following code:

```
1 from kafka import KafkaProducer
2 from configs import kafka_config
3 import json
4 import time
5 import random
6
7 producer = KafkaProducer(
8     bootstrap_servers=kafka_config['bootstrap_servers'],
9     security_protocol=kafka_config['security_protocol'],
10    sasl_mechanism=kafka_config['sasl_mechanism'],
11    sasl_plain_username=kafka_config['username'],
12    sasl_plain_password=kafka_config['password'],
13    value_serializer=lambda v: json.dumps(v).encode('utf-8')
14)
15
16 topic_name = 'Ivan_building_sensors'
17
18 for i in range(100): # Generate data indefinitely
19     data = {
20         "sensor_id": random.randint(1, 100),
21         "timestamp": int(time.time()),
22         "temperature": random.uniform(25, 45),
23         "humidity": random.uniform(15, 85)
24     }
25     producer.send(topic_name, value=data)
26     producer.flush()
27     print(f"Data sent: {data}")
28     time.sleep(1)
29
30 producer.close()
```

The right sidebar shows the Variable Explorer with the `producer` object:

```
producer: producer.kafka.KafkaProducer 1
topic_name: str 21 Ivan_building_sensors
```

The Console window at the bottom shows the output of the script:

```
Topic 'Ivan_humidity_alerts' created successfully.

In [2]: %runfile C:/Work/GoIT/DataEngineering/goit-de-hw-05/producer.py --wdir
Reloaded modules: configs, numpy._utils._conversions, numpy._utils, numpy._globals, numpy._expired_attrs_2_0, numpy.version, numpy._distributor_init, numpy._utils._inspect
Data sent: {'sensor_id': 37, 'timestamp': 1733411115, 'temperature': 28.994102465688425, 'humidity': 66.55663279465489}
Data sent: {'sensor_id': 53, 'timestamp': 1733411115, 'temperature': 39.14672183961571, 'humidity': 16.43977368033383}
Data sent: {'sensor_id': 83, 'timestamp': 1733411116, 'temperature': 36.55175226426349, 'humidity': 40.975534858803876}
Data sent: {'sensor_id': 11, 'timestamp': 1733411117, 'temperature': 26.93221867952238, 'humidity': 41.881423173745235}
Data sent: {'sensor_id': 12, 'timestamp': 1733411118, 'temperature': 41.2096342136831, 'humidity': 37.4819871806937}
Data sent: {'sensor_id': 39, 'timestamp': 1733411119, 'temperature': 35.27945288268208, 'humidity': 65.89876051742363}
Data sent: {'sensor_id': 85, 'timestamp': 1733411120, 'temperature': 33.767660301903284, 'humidity': 23.068414478596587}
Data sent: {'sensor_id': 77, 'timestamp': 1733411121, 'temperature': 38.89347807972018, 'humidity': 20.082759030158037}
Data sent: {'sensor_id': 68, 'timestamp': 1733411122, 'temperature': 27.53881349733547, 'humidity': 74.6222451774676}
Data sent: {'sensor_id': 74, 'timestamp': 1733411123, 'temperature': 34.97548977344858, 'humidity': 27.885962608941232}
Data sent: {'sensor_id': 88, 'timestamp': 1733411124, 'temperature': 44.41915433362159, 'humidity': 79.86579121612255}
Data sent: {'sensor_id': 87, 'timestamp': 1733411125, 'temperature': 44.762171291545506, 'humidity': 59.24418866375885}
Data sent: {'sensor_id': 72, 'timestamp': 1733411126, 'temperature': 41.16106108706609, 'humidity': 28.93225754367318}
Data sent: {'sensor_id': 11, 'timestamp': 1733411127, 'temperature': 26.02618582987982, 'humidity': 47.766231732592416}
Data sent: {'sensor_id': 88, 'timestamp': 1733411128, 'temperature': 26.892524216765018, 'humidity': 21.392652754746}
Data sent: {'sensor_id': 8, 'timestamp': 1733411129, 'temperature': 44.56146519981874, 'humidity': 21.201817884234046}
```

3. Отримання та відправка відфільтрованих даних

The screenshot shows the Spyder Python IDE with the following components:

- Editor:** Contains the code for `consumer.py` and `alerts.py`. The code defines Kafka consumers for temperature and humidity alerts and a producer for sending these alerts.
- Variable Explorer:** Displays the variables in the current scope, including `data`, `i`, `kafka_config`, `producer`, and `topic_name`.
- Console I/O:** Shows the output of the program, including the received data and the sent alerts.

```
1 from kafka import KafkaConsumer, KafkaProducer
2 import json
3 from configs import kafka_config
4
5 consumer = KafkaConsumer(
6     'Ivan_building_sensors',
7     bootstrap_servers=kafka_config['bootstrap_servers'],
8     security_protocol=kafka_config['security_protocol'],
9     sasl_mechanism=kafka_config['sasl_mechanism'],
10    sasl_plain_username=kafka_config['username'],
11    sasl_plain_password=kafka_config['password'],
12    value_deserializer=lambda v: json.loads(v.decode('utf-8')),
13    auto_offset_reset='earliest',
14    enable_auto_commit=True,
15    group_id='sensor_alerts_processor'
16)
17
18 producer = KafkaProducer(
19     bootstrap_servers=kafka_config['bootstrap_servers'],
20     security_protocol=kafka_config['security_protocol'],
21     sasl_mechanism=kafka_config['sasl_mechanism'],
22     sasl_plain_username=kafka_config['username'],
23     sasl_plain_password=kafka_config['password'],
24     value_serializer=lambda v: json.dumps(v).encode('utf-8')
25)
26
27 for message in consumer:
28     data = message.value
29     print("Received data:", data) # Вивід отриманих даних
30
31     if data['temperature'] > 40:
32         alert_temp = {"sensor_id": data['sensor_id'], "alert":
33             "High temperature"}
34         producer.send('Ivan_temperature_alerts', value=alert_temp)
35         print("Sent to Ivan_temperature_alerts:", alert_temp)
36
37     if data['humidity'] > 80 or data['humidity'] < 20:
38         alert_hum = {"sensor_id": data['sensor_id'], "alert":
39             "Abnormal humidity"}
40         producer.send('Ivan_humidity_alerts', value=alert_hum)
41         print("Sent to Ivan_humidity_alerts:", alert_hum)
42
43     producer.flush()
44
45 consumer.close()
46 producer.close()
```

Console I/O output:

```
In [1]: %runfile C:/Work/GoIT/DataEngineering/goit-de-hw-05/consumer.py --wdir
Reloaded modules: configs, numpy, numpy._utils._convertions, numpy._utils._inspect,
numpy._utils._inspect, numpy._utils._inspect, numpy._utils._inspect, numpy._utils._inspect,
Received data: {'sensor_id': 36, 'timestamp': 1733415235, 'temperature': 39.830895625695504, 'humidity':
84.31774169722}
Sent to Ivan_humidity_alerts: {'sensor_id': 36, 'alert': 'Abnormal humidity'}
Received data: {'sensor_id': 9, 'timestamp': 1733415241, 'temperature': 25.405827107573375, 'humidity':
48.54063240073551}
Received data: {'sensor_id': 11, 'timestamp': 1733415246, 'temperature': 33.61454832213693, 'humidity':
48.14848181794592}
Received data: {'sensor_id': 1, 'timestamp': 1733415247, 'temperature': 44.182012942729706, 'humidity':
31.493484305692515}
Sent to Ivan_temperature_alerts: {'sensor_id': 1, 'alert': 'High temperature'}
Received data: {'sensor_id': 3, 'timestamp': 1733415256, 'temperature': 31.853230569455658, 'humidity':
65.50540290705237}
Received data: {'sensor_id': 74, 'timestamp': 1733415259, 'temperature': 30.351809275839336, 'humidity':
61.40045884872029}
Received data: {'sensor_id': 54, 'timestamp': 1733415260, 'temperature': 39.76302907393777, 'humidity':
53.90798371537557}
Received data: {'sensor_id': 81, 'timestamp': 1733415265, 'temperature': 42.56595883464928, 'humidity':
61.67049350727712}
```

4. Результат запису відфільтрованих даних

The screenshot shows the Spyder Python IDE with the following components:

- Editor:** Contains the code for `alerts.py`. The code defines Kafka consumers for temperature and humidity alerts and a producer for sending these alerts.
- Variable Explorer:** Displays the variables in the current scope, including `data`, `i`, `kafka_config`, `producer`, and `topic_name`.
- Console I/O:** Shows the output of the program, including the received data and the sent alerts.

```
1 from kafka import KafkaConsumer
2 import json
3
4 consumer_temp = KafkaConsumer(
5     'Ivan_temperature_alerts',
6     bootstrap_servers=[kafka_config['bootstrap_servers']],
7     auto_offset_reset='earliest',
8     value_deserializer=lambda x: json.loads(x.decode('utf-8'))
9)
10
11 consumer_humidity = KafkaConsumer(
12     'Ivan_humidity_alerts',
13     bootstrap_servers=[kafka_config['bootstrap_servers']],
14     auto_offset_reset='earliest',
15     value_deserializer=lambda x: json.loads(x.decode('utf-8'))
16)
17
18 for message in consumer_temp:
19     print("Temperature Alert:", message.value)
20
21 for message in consumer_humidity:
22     print("Humidity Alert:", message.value)
23
```

Console I/O output:

```
In [1]: %runfile C:/Work/GoIT/DataEngineering/goit-de-hw-05/alerts.py --wdir
Reloaded modules: numpy, numpy._utils._convertions, numpy._utils._inspect,
numpy._utils._inspect, numpy._utils._inspect, numpy._utils._inspect, numpy._utils._inspect,
Temperature Alert: {'sensor_id': 72, 'alert': 'High temperature'}
Temperature Alert: {'sensor_id': 31, 'alert': 'High temperature'}
Temperature Alert: {'sensor_id': 5, 'alert': 'High temperature'}
Temperature Alert: {'sensor_id': 24, 'alert': 'High temperature'}
Temperature Alert: {'sensor_id': 18, 'alert': 'High temperature'}
Temperature Alert: {'sensor_id': 35, 'alert': 'High temperature'}
Temperature Alert: {'sensor_id': 36, 'alert': 'High temperature'}
Temperature Alert: {'sensor_id': 25, 'alert': 'High temperature'}
Temperature Alert: {'sensor_id': 19, 'alert': 'High temperature'}
Temperature Alert: {'sensor_id': 70, 'alert': 'High temperature'}
Temperature Alert: {'sensor_id': 19, 'alert': 'High temperature'}
Temperature Alert: {'sensor_id': 7, 'alert': 'High temperature'}
Temperature Alert: {'sensor_id': 33, 'alert': 'High temperature'}
Temperature Alert: {'sensor_id': 56, 'alert': 'High temperature'}
Temperature Alert: {'sensor_id': 84, 'alert': 'High temperature'}
Temperature Alert: {'sensor_id': 82, 'alert': 'High temperature'}
Temperature Alert: {'sensor_id': 12, 'alert': 'High temperature'}
```

```
C:\Windows\system32\cmd.exe X + v
humidity_alerts
temperature_alerts

C:\Work\kafka_2.12-3.9.0\bin\windows>kafka-console-consumer --bootstrap-server localhost:9092 --topic Ivan_temperature_a
lerts --from-beginning
{"sensor_id": 72, "alert": "High temperature"}
{"sensor_id": 31, "alert": "High temperature"}
{"sensor_id": 5, "alert": "High temperature"}
{"sensor_id": 24, "alert": "High temperature"}
{"sensor_id": 10, "alert": "High temperature"}
{"sensor_id": 35, "alert": "High temperature"}
{"sensor_id": 36, "alert": "High temperature"}
{"sensor_id": 25, "alert": "High temperature"}
{"sensor_id": 19, "alert": "High temperature"}
{"sensor_id": 70, "alert": "High temperature"}
{"sensor_id": 19, "alert": "High temperature"}
{"sensor_id": 7, "alert": "High temperature"}
{"sensor_id": 33, "alert": "High temperature"}
{"sensor_id": 56, "alert": "High temperature"}
{"sensor_id": 84, "alert": "High temperature"}
{"sensor_id": 82, "alert": "High temperature"}
{"sensor_id": 12, "alert": "High temperature"}
{"sensor_id": 88, "alert": "High temperature"}
{"sensor_id": 83, "alert": "High temperature"}
{"sensor_id": 8, "alert": "High temperature"}
{"sensor_id": 3, "alert": "High temperature"}
{"sensor_id": 80, "alert": "High temperature"}
{"sensor_id": 20, "alert": "High temperature"}
{"sensor_id": 13, "alert": "High temperature"}
```

```
C:\Work\kafka_2.12-3.9.0\bin\windows>kafka-console-consumer --bootstrap-server localhost:9092 --topic Ivan_humidity_aler
ts --from-beginning
{"sensor_id": 93, "alert": "Abnormal humidity"}
{"sensor_id": 46, "alert": "Abnormal humidity"}
{"sensor_id": 81, "alert": "Abnormal humidity"}
{"sensor_id": 86, "alert": "Abnormal humidity"}
{"sensor_id": 81, "alert": "Abnormal humidity"}
{"sensor_id": 8, "alert": "Abnormal humidity"}
{"sensor_id": 10, "alert": "Abnormal humidity"}
{"sensor_id": 34, "alert": "Abnormal humidity"}
{"sensor_id": 70, "alert": "Abnormal humidity"}
{"sensor_id": 7, "alert": "Abnormal humidity"}
{"sensor_id": 13, "alert": "Abnormal humidity"}
{"sensor_id": 77, "alert": "Abnormal humidity"}
{"sensor_id": 5, "alert": "Abnormal humidity"}
{"sensor_id": 53, "alert": "Abnormal humidity"}
{"sensor_id": 3, "alert": "Abnormal humidity"}
```