

Introduction to IoT data

ANALYZING IOT DATA IN PYTHON



Matthias Voppichler
IT Developer

Course overview

- Collect and analyze IoT data
- Gather data
 - API Endpoints
 - Data Streams
- Visualize data
- Combine datasets
- Detect patterns
- ML Model based alerts

What is IoT?

IoT == Internet of Things

- Network of connected devices
- Measure and collect data
- Interact with environment

IoT Devices

Connected devices

- Smart locks
- Connected thermostats
- Temperature sensors



Industrial connected devices

- Connected machines
- Robots / Cobots
- Package tracking



IoT Data formats

- http / json
- plain text
- binary data
- XML
- Proprietary protocols

Data acquisition

- Data streams
- Gathered from a device
- API endpoints

Data acquisition - requests

```
import requests
url = "https://demo.datacamp.com/api/temp?count=3"
r = requests.get(URL)
print(r.json())
```

```
[{'timestamp': 1536924000000, 'value': 22.3},
 {'timestamp': 1536924600000, 'value': 22.8},
 {'timestamp': 1536925200000, 'value': 23.3}]
```

```
print(pd.DataFrame(r.json()).head())
```

	timestamp	value
0	1536924000000	22.3
1	1536924600000	22.8
2	1536925200000	23.3

Data acquisition - pandas

```
import pandas as pd
df_env = pd.read_json("https://demo.datacamp.com/api/temp?count=3")
print(df_env.head())
```

```
      timestamp  value
0 2018-09-14 11:20:00  22.3
1 2018-09-14 11:30:00  22.8
2 2018-09-14 11:40:00  23.3
```

```
print(df_env.dtypes)
```

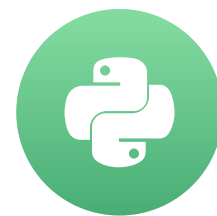
```
timestamp    datetime64[ns]
value        float64
dtype: object
```


Let's Practice

ANALYZING IOT DATA IN PYTHON

Understand the data

ANALYZING IOT DATA IN PYTHON



Matthias Voppichler
IT Developer

Store data to disk

Reasons to store IoT Data

- Limited historical data availability
- Reproducible results
- Training ML Models

Store data using pandas

```
df_env.to_json("data.json", orient="records")
```

```
!cat data.json  
[{'timestamp': 1536924000000, 'value': 22.3},  
 {'timestamp': 1536924600000, 'value': 22.8},  
 {'timestamp': 1536925200000, 'value': 23.3},  
 {'timestamp': 1536925800000, 'value': 23.6},  
 {'timestamp': 1536926400000, 'value': 23.5}]
```

Reading stored data

- From JSON files

```
import pandas
df_env = pd.read_json("data.json")
```

- From CSV file

```
import pandas
df_env = pd.read_csv("data.csv")
```

Validate data load

- Correct column headers
- Check Data formats

```
df_env.head()
```

	timestamp	humidity	pressure	sunshine	temperature
0	2018-09-01 00:00:00	95.6	1016.3	599.2	16.1
2	2018-09-01 00:10:00	95.5	1016.4	600.0	16.1
4	2018-09-01 00:20:00	95.2	1016.5	598.9	16.1
6	2018-09-01 00:30:00	95.1	1016.4	600.0	16.1
8	2018-09-01 00:40:00	95.3	1016.3	600.0	16.1

dataframe.info()

```
df_env.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13085 entries, 0 to 13085
Data columns (total 5 columns):
pressure                13085 non-null float64
humidity                13085 non-null float64
sunshine                13083 non-null float64
temperature             13059 non-null float64
timestamp               13085 non-null datetime64[ns]
dtypes: datetime64[ns](1), float64(4)
memory usage: 1.4 MB
```

pandas describe()

```
df_env.describe()
```

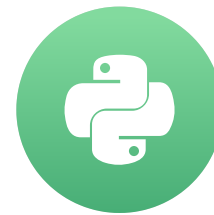
	humidity	pressure	sunshine	temperature
count	13057.000000	13057.000000	13057.000000	13057.000000
mean	73.748350	1019.173003	187.794746	14.06647
std	20.233558	6.708031	274.094951	6.61272
min	8.900000	989.500000	0.000000	-1.80000
25%	57.500000	1016.000000	0.000000	9.80000
50%	78.800000	1019.700000	0.000000	13.40000
75%	91.300000	1023.300000	598.900000	18.90000
max	100.100000	1039.800000	600.000000	30.40000

Time for Practice!

ANALYZING IOT DATA IN PYTHON

Introduction to Data streams

ANALYZING IOT DATA IN PYTHON



Matthias Voppichler
IT Developer

What is a Data Stream

- Constant stream of Data
- Examples
 - Twitter messages
 - Online News Articles
 - Video streams
 - Sensor data (IoT)
 - Market orders (financial)

What is a Data Stream

- Constant stream of Data
- Examples
 - Twitter messages
 - Online News Articles
 - Video streams
 - **Sensor data (IoT)**
 - Market orders (financial)

MQTT

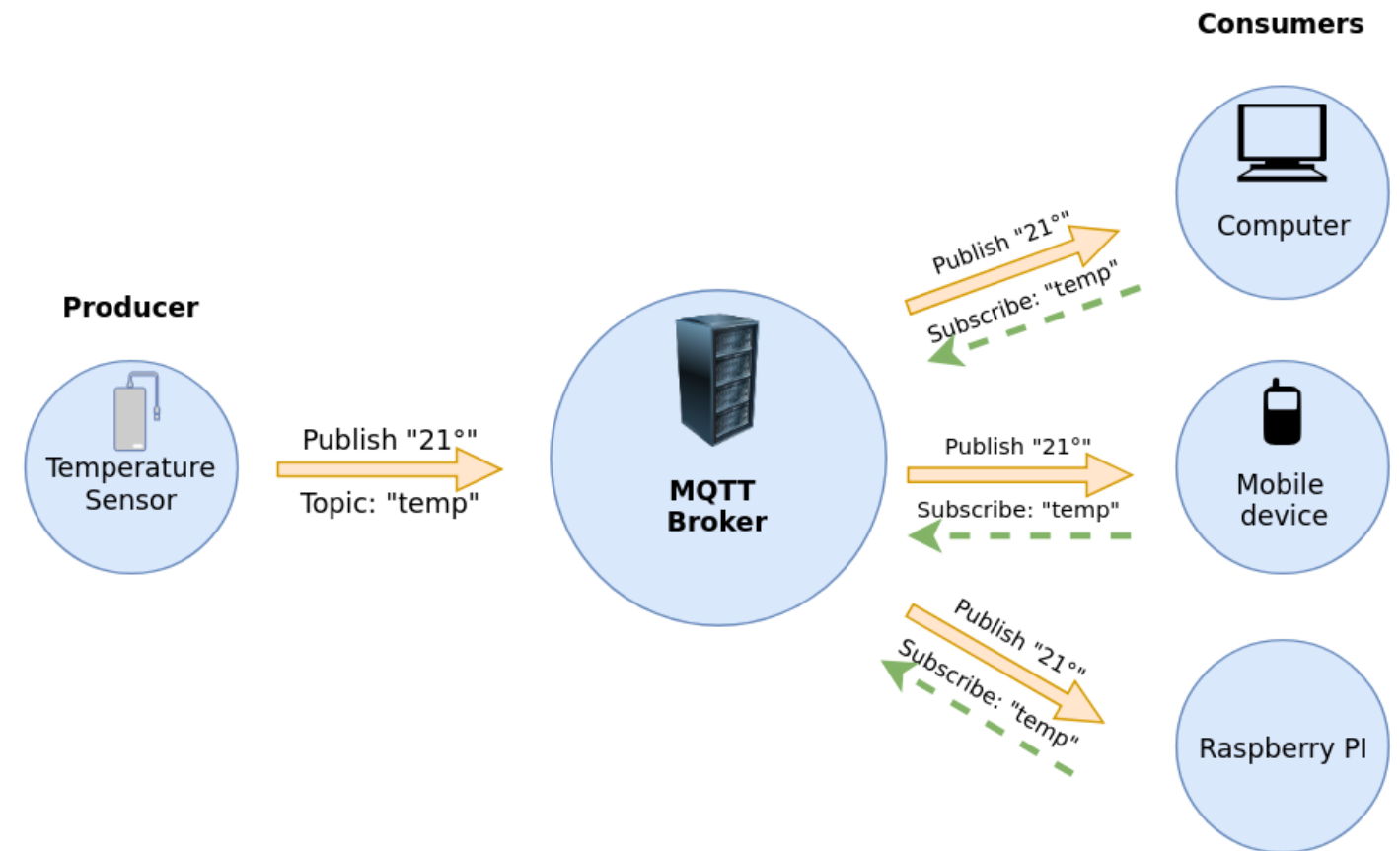
- Message protocol
- Publish / subscribe
- Small footprint

Server -> Acts as a message Broker

Client:

- Connects to a Broker
- Publishes data
- Subscribes to topics

Message Queuing Telemetry Transport



Python library

Eclipse Paho™ MQTT Python Client

```
# Import MQTT library  
import paho.mqtt
```

More information and the documentation available at GitHub

<https://github.com/eclipse/paho.mqtt.python>

Single message

```
import paho.mqtt.subscribe as subscribe
msg = subscribe.simple("paho/test/simple",
                      hostname="test.mosquitto.org")

print(f"{msg.topic}, {msg.payload}")
```

Output:

```
paho/test/simple, {"time": 1549481572, "humidity": 77, "temp": 21}
```

Callback

```
def on_message(client, userdata, message):  
    print(f"{message.topic} : {message.payload}")
```

Arguments

- `client` - client instance
- `userdata` - private user data
- `message` - instance of MQTTMessage

Callback

```
import paho.mqtt.subscribe as subscribe  
  
subscribe.callback(on_message,  
                   topics="datacamp/roomtemp",  
                   hostname="test.mosquitto.org")
```

MQTT Subscribe

```
import paho.mqtt.subscribe as subscribe

def on_message(client, userdata, message):
    print("{} : {}".format(message.topic, message.payload))

subscribe.callback(on_message,
                   topics="datacamp/roomtemp",
                   hostname="test.mosquitto.org")
```

```
datacamp/roomtemp : b'{"time": 1543344857, "hum": 34, "temp": 24}'
datacamp/roomtemp : b'{"time": 1543344858, "hum": 35, "temp": 23}'
datacamp/roomtemp : b'{"time": 1543344860, "hum": 36, "temp": 22}'
datacamp/roomtemp : b'{"time": 1543344946, "hum": 37, "temp": 22}'
datacamp/roomtemp : b'{"time": 1543345010, "hum": 36, "temp": 13}'
```

Let's practice!

ANALYZING IOT DATA IN PYTHON