

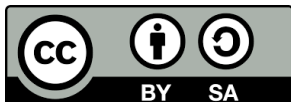
# Sensori di allarme

## Esercizio con soluzioni

### ***Autore(i)/Organizzazione(i):***

Anders Östman (Novogit AB)

### ***Licenza***



<https://creativecommons.org/licenses/by-sa/4.0/>

### ***Versione***

Versione 1.0

Data: Aprile 2024

### ***Risultati di apprendimento***

Al termine di questo esercizio, il discente dovrà essere in grado di:

- Accedere alle misure dei sensori in una rete di sensori.
- Sviluppare un piccolo script Python per gestire le misurazioni dei sensori.

### ***Requisiti in entrata***

- Conoscenza di base di Python.
- Conoscenza di base delle reti di sensori.
- Conoscenza di base del formato JSON



### **Sintesi**

Questo esercizio riguarda la lettura dei dati dei sensori di una rete di sensori per la qualità dell'aria. La rete di sensori è accessibile utilizzando lo standard OGC Sensor Observation (SOS). Se l'inquinamento atmosferico supera una certa soglia, viene emesso un allarme.

### **Carico di lavoro previsto**

Un compito in Python. 8 ore di lezione e compiti a casa, 0,8 ECTS (ECVET)

*Finanziato dall'Unione europea. Le opinioni espresse appartengono, tuttavia, al solo o ai soli autori e non riflettono necessariamente le opinioni dell'Unione europea o dell'Agenzia esecutiva europea per l'istruzione e la cultura (EACEA). Né l'Unione europea né l'EACEA possono esserne ritenute responsabili.*



## Contenuto della lezione

Esercizio	4
Preparazione	5
Suggerimenti	6
Soluzione	7
Fonti	8



### **Esercizio:**

I dati sulla qualità dell'aria sono costantemente monitorati attraverso una rete di stazioni di monitoraggio. Nel contesto svedese, l'Agenzia svedese per la protezione dell'ambiente è responsabile della comunicazione dei dati sulla qualità dell'aria all'Agenzia europea per l'ambiente. Le operazioni delle stazioni di monitoraggio svedesi sono subappaltate a diversi host di dati. Tutte le osservazioni sulla qualità dell'aria sono gestite dall'Istituto meteorologico e idrologico svedese (SMHI).

I dati sulla qualità dell'aria vengono inseriti ogni ora o ogni giorno e resi disponibili attraverso vari servizi web e API. Questo incarico riguarda in particolare le misurazioni del PM<sub>2,5</sub> (aerosol), piccole particelle con un diametro massimo di 2,5 micrometri. Secondo la norma svedese per l'aria pulita, la concentrazione di PM<sub>2,5</sub> non può superare i 25 microgrammi per metro cubo d'aria, calcolati come valore medio al giorno. Tuttavia, secondo le raccomandazioni dell'Organizzazione Mondiale della Sanità (OMS), che presto saranno anche la norma europea, la media giornaliera non dovrebbe superare i 15 microgrammi per metro cubo d'aria.

Le stazioni che misurano il contenuto di PM<sub>2,5</sub> lo riportano su base oraria. Il vostro compito è quello di sviluppare un software che legga il contenuto di PM<sub>2,5</sub> in una stazione di monitoraggio ed emetta un avviso se il contenuto di PM<sub>2,5</sub> nell'ultima ora registrata supera i 15 microgrammi per metro cubo d'aria. Si noti che l'allarme deve essere emesso su base oraria, anche se la norma sulla qualità dell'aria si riferisce alla media giornaliera.



## Preparazione

Predisporre l'ambiente di sviluppo del software. Se si desidera sviluppare il software in Python, al di fuori dell'ambiente QGIS, è necessario scaricare IDLE.

1. Scaricare e installare la piattaforma di sviluppo Python IDLE da <https://www.python.org/downloads/>.
2. Predisporre l'ambiente Python installando le classi necessarie per accedere alle risorse web. Aprire la console di comando e inserire il seguente comando  
`$ python3 -m pip install requests`

Trovare l'ID della stazione di monitoraggio a cui si desidera accedere.

1. Aprire QGIS e aprire una mappa di sfondo. Per gli utenti svedesi, la mappa topowebb di Lantmateriet può essere un WMS adatto (<https://www.lantmateriet.se/sv/geodata/vara-produkter/produktlista/topografisk-webbkarta-visning/>). È necessario richiedere un token per avere accesso a questa mappa. Un'altra opzione è quella di utilizzare Open Street Map e il plugin QuickOSM come mappa di sfondo.
2. Le posizioni di tutte le stazioni di monitoraggio sono fornite dall'EPA svedese attraverso un WFS. Collegarsi a questo servizio utilizzando l'URL <https://datavardluft.smhi.se/shair/wfs>.
3. Per conoscere l'ID della nostra stazione di monitoraggio, utilizzare il pulsante "Identifica oggetto" e fare clic su una stazione adeguata. Verificare innanzitutto che la stazione misuri il PM2,5, accertandosi che l'attributo `sampling_point_property_notation` sia impostato su PM2,5. Tenere presente che ogni stazione di monitoraggio può avere diversi sensori, quindi potrebbe essere necessario esaminare più di un sensore. L'ID da usare quando si interroga il SOS è dato dall'attributo `samplingpoint_process_inspireid`. Nel mio esempio, ho utilizzato una stazione di monitoraggio a Uppsala, con l'ID "SPP-SE159404\_06001\_100\_100".
4. Le REST-API sono in formato json, mentre le SOS sono originariamente in XML. XML offre una maggiore flessibilità, ma è più facile lavorare con json. Fortunatamente, l'SOS che stiamo utilizzando è in grado di rispondere anche in formato json, il che semplifica il nostro compito.

Aprire la piattaforma di sviluppo Python e iniziate a lavorare sul compito.



## Suggerimenti:

- Suggerimento 1:** Un problema nel trattare i dati esterni è scoprire il significato dei termini utilizzati. Nell'esempio svedese, alcuni suggerimenti sono riportati sopra nel testo di preparazione. Altrimenti, potrebbe essere difficile trovare il significato degli altri termini utilizzati. I termini più generali sono specificati negli standard OGC, in questo caso l'OGC SOS Interface Standard.
- Suggerimento 2:** importare anche l'estensione json. Ciò consente di lavorare con i file json in modo pratico.
- Suggerimento 3:** eseguire stampe durante lo sviluppo, in modo da poter vedere i dati con cui si lavora. Usare ad esempio `json.dump` per formattare i dati come json.



## Soluzione normativa

```
# Reading sensor observations

import requests
import json

sppInspireId = "SPP-SE159404_06001_100_100"
Threshold = 15

print ("Reading SMHI sensor observations - SOS standard")
sosEndpoint = "https://datavardluft.smhi.se/52North/service?"
sosCommon = "service=SOS&version=2.0.0&"
getObservationString = "REQUEST=GetObservation&procedure=" + sppInspireId + "&responseFormat=application/json"

# Get all observations for this station
print ("Getting Observations")
outputfile = 'Observations.json'
sosUrl = sosEndpoint + sosCommon + getObservationString
response = requests.get(sosUrl)
json_formatted_str = json.dumps(response.json(), indent=2)
print(json_formatted_str,file=open(outputfile, 'w'))

# Find last observation and print date/time and value
print ("Extracting most recent observation")
Obs = response.json()["observations"] # Strip off the header information
lastObs = Obs[0]["phenomenonTime"][1] # Last observation is the first one

for sensorObs in Obs : # Check if later observations
    thisObsTime = sensorObs["phenomenonTime"][1]
    if thisObsTime > lastObs :
        lastObs = thisObsTime
        lastValue = sensorObs["result"]["value"]

print ("\n\nMost recent observation")
print ("Time: ", lastObs, "\nValue: ", lastValue, "\n\n")

if lastValue > Threshold :
    print ("\nWARNING: Threshold exceeded")
else :
    print ("\nAir Quality OK")

print ("Reading sensor observations finished")
```



## Fonti

General description of air quality data and reporting

<https://www.naturvardsverket.se/amnesomraden/luft/statistik--utslapp-och-halter/luftkvaliteten-i-realtid-och-preliminar-statistik/>

Description of sensor data and access.

<https://www.naturvardsverket.se/amnesomraden/luft/statistik--utslapp-och-halter/luftkvaliteten-i-realtid-och-preliminar-statistik/webbtjanster-luftkvalitetsdata/>

OGC, Sensor Observation Service Standard. <https://www.ogc.org/standard/sos/>

Link to GeoCOVID Watch Sensor Things API: <http://covidsta.hft-stuttgart.de/server/v1.1>

Basics of REST API programming. <https://realpython.com/api-integration-in-python/>

<https://dd.eionet.europa.eu/vocabularyconcept/aq/pollutant/5/view>

<https://realpython.com/api-integration-in-python/>