

Sistemi e Applicazioni Cloud - Laboratorio

Appello 16 gennaio 2023 [Tempo consegna: 2h 30m]

Si realizzi un'applicazione su Google Cloud Platform per la gestione delle bollette di una generica utility legata al consumo di energia elettrica. Si vogliono gestire le letture dei consumi e l'emissione di bollette

Si rendano disponibili i seguenti tipi di interfacce di utilizzo:

1. API per inserire le letture
2. Pagina Web per la visualizzazione delle bollette

L'applicazione deve essere testata per il deployment su piattaforma GCP utilizzando i servizi visti a lezione:

- App Engine
- Firestore
- Functions

Parte 1 - Backend API REST

L'applicazione deve esporre le seguenti funzionalità tramite opportune Web API RESTful:

1. inviando richieste **POST** allo URI `/api/v1/consumi/{data}` è possibile inserire una nuova lettura.
2. inviando richieste **GET** allo URI `/api/v1/consumi/{data}` è possibile leggere il valore immesso oppure una stima dei consumi a quella data
3. inviando richieste **POST** allo URI `/api/v1/clean` è possibile cancellare i consumi memorizzati nel database

L'interfaccia di utilizzo delle API deve soddisfare **rigorosamente** il file di specifica *OpenAPI* disponibile insieme alla presente specifica

Per testare la funzionalità dell'API è possibile far riferimento al seguente URL:

`https://sacvalidator.appspot.com/`

Parte 2 - Web Application

Realizzare due pagine Web:

1. Una pagina con le date delle ultime 12 bollette (emesse il primo giorno del mese)
2. Una pagina con il dettaglio dei consumi di una bolletta

La pagina con la lista delle bollette deve avere i link per accedere alle varie bollette.

La pagina con il dettaglio di una bolletta deve contenere:

- Il periodo di riferimento della bolletta (corrispondente al mese precedente)

- Il costo complessivo della bolletta
- L'ultima lettura considerata
- I consumi del mese

Parte 3 - Interconnessione di servizi

Realizzare un sistema per popolare automaticamente il database delle bollette sulla base delle letture. In particolare si vuole usare una funzione che:

- Abbia come trigger l'inserimento di una lettura dei consumi nel database
- Abbia come azione una funzione che popola un database con i valori per la prossima bolletta

Non è necessario collegare la funzione all'applicazione Web. Basta mostrare che il database con i dati delle bollette si popola.

Note

- Se la validazione dell'API con il validatore non è positiva, l'esame è automaticamente insufficiente.
- In caso di validazione positiva il voto sarà basato su quante parti dell'esame sono state portate a termine e sulla qualità del codice e delle soluzioni proposte.

Dettagli implementativi

Per l'implementazione si ricorda che:

- Per memorizzare le letture dei consumi si può usare il database Firestore con una collezione *letture* in cui ciascun documento ha come nome la data della lettura.
- Per memorizzare bollette si può usare una collezione *bollette* in cui ciascun documento ha un nome che contiene il mese e l'anno della bolletta (ci può essere anche il giorno ma non è strettamente necessario).
- L'unità di misura dell'energia elettrica è il KWh. L'attuale costo dell'energia elettrica è di circa 0.5 euro/KWh. Per semplicità si assume che il costo sia sempre costante.
- Si consiglia di sviluppare una funzione helper che, partendo da una data, ritorni la prima lettura immediatamente precedente a tale data
- Per risparmiare tempo, si suggerisce di implementare sia le API che la applicazione Web in un unico sorgente e di farne il deployment come unico servizio di default
- La differenza tra due date è di tipo *timedelta* che si può convertire in secondi con il metodo *.total_seconds()*
- La funzione che stima i consumi è fornita nel seguito.
- Le date sono in formato giorno-mese-anno. Per le conversioni si possono usare le funzioni riportate nel seguito

Funzione di interpolazione dei consumi

Si vuole calcolare il valore dei consumi c_x al tempo t_x .

Se sono presenti almeno due letture nel database precedenti a t_x si considerano le ultime due letture. Sia c_1 al tempo t_1 e c_2 al tempo t_2 , con $t_1 < t_2 < t_x$.

$$c_x = c_2 + \frac{c_2 - c_1}{t_2 - t_1}(t_x - t_2)$$

Se è presente una sola lettura c_1 precedente a t_x si considera che i consumi non siano variati dall'ultima lettura. Quindi Il valore al tempo t_2 diventa:

$$c_x = c_1$$

Se non è presente nessun valore, il risultato è semplicemente $c_x = 0$

Funzione per gestire le date

```
def date_from_str(d):  
    try: return datetime.strptime(d, '%d-%m-%Y')  
    except: return None  
  
def str_from_date(d):  
    return d.strftime('%d-%m-%Y')
```