 BigProfiles	DOCUMENTO IT/HR		IT-10.2021
	TEST TECNICO BACKEND		Rev 01



SEDE LEGALE: VIA COSTANTINO, 73 - 00145 ROMA

SEDE OPERATIVA: VIA MARSALA, 29/H - 00185 ROMA


DOCUMENTO IT/HR

TEST TECNICO BACKEND

Identificazione del documento						
Codice Documento	IT-10.2021	Revisione	01	Livello Riservatezza	Confidenziale	Pag. 1 di 7

<input checked="" type="checkbox"/>	Copia non controllata		
<input type="checkbox"/>	Copia Controllata	Nr.	Trasmessa a:


INDICE DELLE REVISIONI			
REV.	DATA	DESCRIZIONE MODIFICA	Paragrafo
00	28.07.2021	PRIMA VERSIONE	
01	18.10.2022	AGGIORNAMENTO DI ALCUNE SPECIFICHE	

	DOCUMENTO IT/HR	IT-10.2021
	TEST TECNICO BACKEND	Rev 01

Indice

1	Introduzione	4
2	Specifiche	4
	API.....	4
	Middleware	4
	Componenti.....	5
3	Esempio.....	5
4	Operatività.....	6
4.1	Extra	6
5	Consegna.....	6
6	Valutazione	6

Identificazione del documento						
Codice Documento	IT-10.2021	Revisione	01	Livello Riservatezza	Confidenziale	Pag. 3 di 7

	DOCUMENTO IT/HR			IT-10.2021
	TEST TECNICO BACKEND			Rev 01

1 Introduzione

Il candidato dovrà realizzare una API ad alte performance per l'ingestion di informazioni. Sui dati salvati dovrà essere possibile l'aggregazione temporale per minuto (ovvero devo poter mettere insieme tutte le chiamate arrivate al minuto 2 della data 2021-07-29 18:02:00) e dovranno essere restituite alcune statistiche (come, ad esempio, tempo totale di risposta, numero di errori, numero di richieste) aggregate sempre per minuto e la lista delle ultime 10 chiamate effettuate dell'ultima aggregazione.

2 Specifiche

API

Deve essere costruita un API che sia **conforme alle caratteristiche** specificate dentro il file "api.json" (in formato OpenAPI)

Questa API è formata da due endpoint distinti:

- Endpoint "ingestion": usato per andare ad inserire i dati all'interno del sistema
- Endpoint "retrieve": usato per andare ad estrarre i dati aggregati dal sistema

I vari endpoint dovranno essere protetti da un sistema di autenticazione basato su chiave: dovrà essere necessario passare il parametro "x-api-key" all'interno dell'header con valore "BigProfiles-API".

2.1 Ingestion

E' un endpoint in modalità POST che prende in input il seguente formato di dati (in formato JSON):

- "key": chiave numerica che varia da 1 a 6
- "payload": una stringa con lunghezza variabile da 10 a 255

Questi dati vengono presi in input e salvati su un database MongoDB, su cui poi andranno effettuate le varie operazioni di aggregazione, conteggi, ecc.

Questa API dovrà avere un comportamento randomico per quanto riguarda il codice di risposta ed il tempo di risposta con le seguenti probabilità:

- Tempo di risposta randomico tra 10ms e 50ms
- Probabilità di errore 10% (errore 500 in risposta)

Sul database dovranno essere salvate le seguenti informazioni:

- key: chiave usata in fase di ingestion
- creation_datetime: datetime di arrivo della richiesta
- payload: payload della richiesta
- response_time: tempo della risposta
- response_code: codice della risposta

E' possibile andare ad utilizzare la forma che più si ritiene opportuna, basta che siano presenti le informazioni precedentemente illustrate.

2.2 Retrieve

E' un endpoint in modalità GET che prende in input il seguente formato di dati:

Identificazione del documento						
Codice Documento	IT-10.2021	Revisione	01	Livello Riservatezza	Confidenziale	Pag. 4 di 7

- date_from: data di inizio della finestra temporale di selezione (in formato ISO)
- date_to: data di fine della finestra temporale di selezione (in formato ISO)

In uscita andranno mandate diverse statistiche, raggruppate per minuto e relative alla finestra temporale scelta, quali numero di richieste effettuate, numero di errori, tempo totale delle risposte e la lista degli ultimi 10 log dell'ultima aggregazione nella finestra temporale.

Nel file "api.json" viene mostrata in modo migliore la struttura che ci si aspetta in uscita.

Esempio

Ad esempio, vediamo che vengono mandate 10 richieste di ingestion con i seguenti valori:

Key	Payload	Response Time	Response Code	Time
1	Stringa Esempio	23	200	2021-07-28 11:01:12.123
2	Stringa Esempio	12	200	2021-07-28 11:01:12.124
3	Stringa Esempio	43	200	2021-07-28 11:01:13.125
1	Stringa Esempio	23	200	2021-07-28 11:01:14.123
1	Stringa Esempio	14	500	2021-07-28 11:02:13.424
2	Stringa Esempio	50	200	2021-07-28 11:02:13.429
3	Stringa Esempio	43	200	2021-07-28 11:02:14.429
2	Stringa Esempio	32	200	2021-07-28 11:02:15.429
2	Stringa Esempio	39	200	2021-07-28 11:02:17.429
1	Stringa Esempio	14	200	2021-07-28 11:02:20.429

Sull'endpoint retrieve, usando come finestra temporale 2021-07-28 11:01:00 - 2021-07-28 11:02:00 usciranno i seguenti valori

key	total_response_time_ms	total_requests	total_errors	creation_datetime
1	46	2	0	2021-07-28 11:01:00
2	12	1	0	2021-07-28 11:01:00
3	43 Esempio	1	0	2021-07-28 11:01:00
1	28	2	1	2021-07-28 11:02:00
2	121	3	0	2021-07-28 11:02:00
3	43	1	0	2021-07-28 11:02:00

Per quanto riguarda i log invece abbiamo i seguenti valori

key	payload	Response_time	Response_code	creation_datetime
-----	---------	---------------	---------------	-------------------

1	Stringa Esempio	14	200	2021-07-28 11:02:20.429
2	Stringa Esempio	39	200	2021-07-28 11:02:17.429
2	Stringa Esempio	32	200	2021-07-28 11:02:15.429
3	Stringa Esempio	43	200	2021-07-28 11:02:14.429
2	Stringa Esempio	50	200	2021-07-28 11:02:13.429
1	Stringa Esempio	14	500	2021-07-28 11:02:13.424

3 Operatività

Il test si può dividere in 3 punti (il candidato può decidere se fare tutti i punti o no):

- 1) Costruzione delle api di ingestion (senza la parte di salvataggio sul database).
- 2) Inserire il salvataggio delle informazioni su database
- 3) Costruzione dell'endpoint retrieve

3.1 Requisiti

- Deve essere usato Python 3.7+
- E' possibile usare un framework a scelta tra FastAPI(congiato)/Flask
- Il salvataggio dei dati deve essere fatto su MongoDB
- Il servizio dovrà esporre la porta operativa 8000

3.2 Extra

Verrà valutato positivamente l'utilizzo di Docker per quanto riguarda il deploy dell'applicativo. Il modulo può essere inserito all'interno del docker-compose fornito.

Per lanciare lo stack docker è presente all'interno della cartella "bin" lo script "start.sh"; per spegnere lo stack usare lo script "stop.sh".


4 Consegna

La consegna deve avvenire al più una settimana dopo dalla consegna del test. Deve essere creata un repository Github con tutto il codice prodotto e fornito una **documentazione** in cui vengono mostrate le scelte effettuate implementative (con attenzione agli aspetti di performance) e i comandi per lanciare l'applicativo stesso

5 Valutazione

Il test verrà valutato secondo i seguenti criteri:

- Correttezza del codice generato
- Qualità del codice stesso (organizzazione, commenti, test, ecc)
- Performance generali

	DOCUMENTO IT/HR	IT-10.2021
	TEST TECNICO BACKEND	Rev 01

Le performance verranno valutate andando ad usare uno script che effettuare **10K chiamate su 10 thread paralleli** (quindi 100K chiamate totali), le performance attese sono di un utilizzo di CPU su singolo di core sotto al 40% ed un utilizzo di memoria sotto al di sotto dei 500MB ed un tempo di esecuzione < 45sec

6 Tips and Tricks

- Utilizzate una forma di salvataggio che permetta di avere ottime performance sia in scrittura delle informazioni che per la fase di retrieve: la forma può essere complicata quanto si vuole, basta che le informazioni in uscita siano corrette
- Poiché si tratta di un'applicazione con un alto flusso di chiamate API bisogna gestire in modo ottimizzato il salvataggio su Database.

Identificazione del documento						
Codice Documento	IT-10.2021	Revisione	01	Livello Riservatezza	Confidenziale	Pag. 7 di 7