



# Documentazione API

## Informazioni sul Documento

<b>Versione</b>	2.0.0
<b>Data di approvazione</b>	2022-06-18
<b>Approvatori</b>	Pietro Villatora
<b>Redattori</b>	Michele Gatto
<b>Verificatori</b>	Francesco Protopapa
<b>Uso</b>	Esterno
<b>Distribuzione</b>	Prof. Vardanega Tullio Prof. Cardin Riccardo Zero12 Gruppo <i>Dream Team</i>

*e-mail: dreamteam.unipd@gmail.com*



## Registro delle Modifiche

Versione	Data	Nominativo	Ruolo	Descrizione
v2.0.0	2022-06-18	Pietro Villatora	Progettista	Approvazione del documento.
v1.1.0	2022-06-16	Michele Gatto	Progettista	Verifica di coesione e consistenza; (Verificatore: <i>Francesco Protopapa</i> )
v1.0.1	2022-06-15	Michele Gatto	Progettista	Aggiunte §2.5, §2.8, §2.7; (Verificatore: <i>Francesco Protopapa</i> )
v1.0.0	2022-05-13	Pietro Villatora	Progettista	Approvazione del documento
v0.1.0	2022-05-13	Michele Gatto	Progettista	Verifica di coesione e consistenza; (Verificatore: <i>Francesco Protopapa</i> )
v0.0.3	2022-05-13	Michele Gatto	Progettista	Stesura §3; (Verificatore: <i>Francesco Protopapa</i> )
v0.0.2	2022-05-12	Michele Gatto	Progettista	Stesura §2; (Verificatore: <i>Francesco Protopapa</i> )
v0.0.1	2022-05-11	Michele Gatto	Amministratore	Creazione scheletro documento e stesura §1; (Verificatore: <i>Francesco Protopapa</i> )



## Indice

<b>1</b>	<b>Introduzione</b>	<b>5</b>
1.1	Scopo del Documento . . . . .	5
1.2	Scopo del Prodotto . . . . .	5
1.3	Glossario . . . . .	5
<b>2</b>	<b>API Ranking Service</b>	<b>6</b>
2.1	GET - /getRanking . . . . .	6
2.1.1	Endpoint . . . . .	6
2.1.2	Parametri . . . . .	6
2.1.3	Esempio di richiesta . . . . .	6
2.1.4	Esempio di risposta . . . . .	7
2.2	POST - /getLabelAndPost . . . . .	8
2.2.1	Endpoint . . . . .	8
2.2.2	Parametri . . . . .	8
2.2.3	Esempio di richiesta . . . . .	8
2.2.4	Esempio di risposta . . . . .	9
2.3	POST - /searchByName . . . . .	11
2.3.1	Endpoint . . . . .	11
2.3.2	Parametri . . . . .	11
2.3.3	Esempio di richiesta . . . . .	11
2.3.4	Esempio di risposta . . . . .	11
2.4	GET - /favorites . . . . .	13
2.4.1	Endpoint . . . . .	13
2.4.2	Parametri . . . . .	13
2.4.2.1	Aggiunta preferito . . . . .	13
2.4.2.2	Rimozione preferito . . . . .	13
2.4.2.3	Restituzione lista preferiti . . . . .	14
2.4.3	Esempio di richiesta . . . . .	14
2.4.3.1	Aggiunta preferito . . . . .	14
2.4.3.2	Rimozione preferito . . . . .	14
2.4.3.3	Restituzione lista preferiti . . . . .	14
2.4.4	Esempio di risposta . . . . .	15
2.4.4.1	Aggiunta preferito . . . . .	15
2.4.4.2	Rimozione preferito . . . . .	15
2.4.4.3	Restituzione lista preferiti . . . . .	15
2.5	GET - /getCities . . . . .	16
2.5.1	Endpoint . . . . .	16
2.5.2	Parametri . . . . .	16
2.5.3	Esempio di richiesta . . . . .	16
2.5.4	Esempio di risposta . . . . .	16
2.6	GET - /getLabelCategoria . . . . .	17
2.6.1	Endpoint . . . . .	17
2.6.2	Parametri . . . . .	17
2.6.3	Esempio di richiesta . . . . .	17
2.6.4	Esempio di risposta . . . . .	17
2.7	GET - /generalFilter . . . . .	18
2.7.1	Endpoint . . . . .	18
2.7.2	Parametri . . . . .	18
2.7.3	Esempio di richiesta . . . . .	18
2.7.4	Esempio di risposta . . . . .	18
<b>3</b>	<b>API Crawling Service</b>	<b>20</b>
3.1	POST - /test_post . . . . .	20
3.1.1	Endpoint . . . . .	20
3.1.2	Parametri . . . . .	20
3.1.3	Esempio di richiesta . . . . .	20
3.1.4	Esempio di risposta . . . . .	20



## Elenco delle tabelle

2	Parametri accettati da <code>/getRanking</code> . . . . .	6
3	Parametri accettati da <code>/getLabelAndPost</code> . . . . .	8
4	Parametri accettati da <code>/searchByName</code> . . . . .	11
5	Parametri accettati da <code>/favorites</code> per l'aggiunta di un preferito . . . . .	13
6	Parametri accettati da <code>/favorites</code> per la rimozione di un preferito . . . . .	13
7	Parametri accettati da <code>/favorites</code> per la restituzione della lista dei preferiti . . . . .	14
8	Parametri accettati da <code>/generalFilter</code> . . . . .	18
9	Parametri accettati da <code>/test_post</code> . . . . .	20

## Listings

1	Esempio di richiesta <code>/getRanking</code> . . . . .	6
2	Esempio di risposta <code>/getRanking</code> . . . . .	7
3	Esempio di richiesta <code>/getLabelAndPost</code> . . . . .	8
4	Esempio di risposta <code>/getLabelAndPost</code> . . . . .	9
5	Esempio di richiesta <code>/searchByName</code> . . . . .	11
6	Esempio di richiesta <code>/searchByName</code> . . . . .	11
7	Esempio di richiesta <code>/favorites</code> per l'aggiunta di un preferito . . . . .	14
8	Esempio di richiesta <code>/favorites</code> per la rimozione di un preferito . . . . .	14
9	Esempio di richiesta <code>/favorites</code> per la restituzione della lista dei preferiti . . . . .	14
10	Esempio di risposta <code>/favorites</code> per l'aggiunta di un preferito . . . . .	15
11	Esempio di risposta <code>/favorites</code> per la rimozione di un preferito . . . . .	15
12	Esempio di risposta <code>/favorites</code> per la restituzione della lista dei preferiti . . . . .	15
13	Esempio di richiesta <code>/getCities</code> . . . . .	16
14	Esempio di richiesta <code>/getCities</code> . . . . .	16
15	Esempio di richiesta <code>/getLabelCategoria</code> . . . . .	17
16	Esempio di richiesta <code>/getLabelCategoria</code> . . . . .	17
17	Esempio di richiesta <code>/searchByName</code> . . . . .	18
18	Esempio di richiesta <code>/getLabelCategoria</code> . . . . .	18
19	Esempio di richiesta <code>/test_post</code> . . . . .	20
20	Esempio di risposta <code>/test_post</code> . . . . .	20



# 1 Introduzione

## 1.1 Scopo del Documento

Lo scopo di questo documento è di fornire una documentazione dettagliata delle API realizzate per il progetto *Sweet*, specificando per ciascuna API i suoi endpoint, le informazioni necessarie ad effettuare una richiesta e la struttura dei messaggi di risposta.

## 1.2 Scopo del Prodotto

L'obiettivo di Sweet e dell'azienda Zero12 è la creazione di un sistema software costituito da una Webapp. Lo scopo del prodotto è di fornire all'utente una guida dei locali gastronomici sfruttando i numerosi contenuti digitali creati dagli utenti sulle principali piattaforme social (Instagram e TikTok). In questo modo, è possibile realizzare una classifica basata sulle impressioni e reazioni di chiunque usufruisca dei servizi dei locali, non solo da professionisti ed esperti del settore.

## 1.3 Glossario

Per evitare ambiguità relative alle terminologie utilizzate è stato creato un documento denominato “*Glossario*”. Questo documento comprende tutti i termini tecnici scelti dai membri del gruppo e utilizzati nei vari documenti con le relative definizioni. Tutti i termini inclusi in questo glossario, vengono segnalati all'interno del documento con l'apice <sup>G</sup> accanto alla parola.

## 2 API Ranking Service

### 2.1 GET - /getRanking

Restituisce una lista di oggetti che rappresentano i ristoranti, ordinati in maniera decrescente relativamente alla somma dei punteggi. Per ogni ristorante vengono fornite le seguenti informazioni:

- L'id univoco;
- Il nome;
- L'indirizzo (opzionale);
- Il numero di telefono (opzionale);
- Il sito web (opzionale);
- La latitudine
- La longitudine
- La categoria
- Il punteggio delle emoji (opzionale);
- Il punteggio delle foto (opzionale);
- Il punteggio del testo (opzionale);
- La prima immagine senza label "Person" (fornita con link presigned con scadenza di un'ora).

#### 2.1.1 Endpoint

`https://cs0thtwbr7.execute-api.eu-central-1.amazonaws.com/dev/getRanking`

#### 2.1.2 Parametri

Nome	Opzionale	Descrizione	Esempio	Default
size	Sì	Rappresenta la quantità di risultati da restituire	20	10
from	Sì	La posizione dalla quale partire per restituire i risultati	5	0

Tabella 2: Parametri accettati da /getRanking

#### 2.1.3 Esempio di richiesta

```
curl -X GET
https://cs0thtwbr7.execute-api.eu-central-1.amazonaws.com/dev/
getRanking?from=5&size=2
```

Listing 1: Esempio di richiesta /getRanking

### 2.1.4 Esempio di risposta

```
[
  {
    "id_ristorante": 984,
    "nome_ristorante": "Lasarte",
    "indirizzo": "",
    "telefono": "+34934453242",
    "sito_web": "http://www.restaurantlasarte.com",
    "latitudine": "41.3937",
    "longitudine": "2.1621",
    "categoria": "Restaurant",
    "punteggio_emoji": 58.0,
    "punteggio_foto": 39.9,
    "punteggio_testo": 64.9,
    "url_image": "<presigned-url>"
  },
  {
    "id_ristorante": 1146,
    "nome_ristorante": "Sublime Hotels",
    "indirizzo": "",
    "telefono": "+351 269449376",
    "sito_web": "http://www.sublimecomporta.pt",
    "latitudine": "38.2809",
    "longitudine": "-8.6990",
    "categoria": "Hotel resort",
    "punteggio_emoji": 48.6077,
    "punteggio_foto": 22.4,
    "punteggio_testo": 91.0,
    "url_image": "<presigned-url>"
  }
]
```

Listing 2: Esempio di risposta /getRanking



## 2.2 POST - /getLabelAndPost

Restituisce una lista contenente tutti i media relativi ad un ristorante insieme a tutte le label riguardanti il cibo disponibili per ogni media restituito. Per ogni media vengono restituite le seguenti informazioni:

- L'id univoco;
- Il nome utente dell'autore;
- La data in cui è stato pubblicato;
- L'id univoco del ristorante al quale è collegato;
- Il testo;
- Il punteggio delle emoji;
- Il punteggio delle immagini;
- Il punteggio del testo;
- La lista delle immagini contenute (fornite con link presigned con scadenza di un'ora).

### 2.2.1 Endpoint

<https://cs0thtwbr7.execute-api.eu-central-1.amazonaws.com/dev/getLabelAndPost>

### 2.2.2 Parametri

Nome	Opzionale	Descrizione	Esempio	Default
id_rist	No	L'id univoco del ristorante	6	

Tabella 3: Parametri accettati da /getLabelAndPost

### 2.2.3 Esempio di richiesta

```
curl -X POST
https://cs0thtwbr7.execute-api.eu-central-1.amazonaws.com/dev/
getLabelAndPost
-H 'Content-Type: application/json'
-d '{"id_rist":"6"}'
```

Listing 3: Esempio di richiesta /getLabelAndPost



## 2.2.4 Esempio di risposta

```
[
  {
    "id_post": "2756724839010375739_261493927",
    "nome_utente": "cavallitos",
    "data_post": "2022-01-22",
    "id_ristorante": 6,
    "testo": "Asador in salsa verde. Tra le altre cose, hamburger
      nippo-piemontese di wagyu e vicciola.",
    "punteggio_emoji": null,
    "punteggio_testo": 28.5,
    "punteggio_foto": null,
    "url_immagine": [
      "<presigned-url>",
      "<presigned-url>",
      "<presigned-url>",
      "<presigned-url>"
    ],
    "nome_label": [
      [
        {
          "nome_label": "Pork"
        }
      ],
      [
        {
          "nome_label": "Sweets"
        }
      ],
      [
        {
          "nome_label": "Bread"
        }
      ],
      [
        {
          "nome_label": "Pork"
        }
      ]
    ]
  },
  {
    "id_post": "2801767514265164419_5450171258",
    "nome_utente": "paolo_vizzari",
    "data_post": "2022-03-25",
    "id_ristorante": 6,
    "testo": "Dovremmo cercare di mangiar carne una volta di meno e
      mangiarla dieci volte meglio. \n \n Roberto e' il mio
      personale sacerdote della bistecca, l'unico capace di
      emozionarmi tre volte per ogni ciccia: quando la racconta,
      mentre la cuoce e colpo finale al morso. Un sardone
      trapiantato a Torino col cuore grande come le sue costate e
      prezioso come le sue frollature perfette.\n\nCi vuol poco a
      bruciare una poesia, molto piu' amore per scriverne i versi
      con la brace.",
    "punteggio_emoji": null,
    "punteggio_testo": 98.0,
```

```
[{"punteggio_foto": 0.8,
  "url_immagine": [
    "<presigned-url>",
    "<presigned-url>",
    "<presigned-url>"
  ],
  "nome_label": [
    [],
    [
      {
        "nome_label": "Steak"
      }
    ],
    []
  ]
}]
```

Listing 4: Esempio di risposta /getLabelAndPost



## 2.3 POST - /searchByName

Restituisce tutte le informazioni (come già definite in §2.1) per il ristorante il cui nome che ha un match esatto con il parametro passato oppure, se il match esatto non viene trovato, ritorna una lista contenente le informazioni di tutti ristoranti i cui nomi contengono, come sotto-stringhe, almeno una delle singole parole passate come parametro.

### 2.3.1 Endpoint

<https://cs0thtwbr7.execute-api.eu-central-1.amazonaws.com/dev/searchByName>

### 2.3.2 Parametri

Nome	Opzionale	Descrizione	Esempio	Default
name	No	Il nome del ristorante da cercare	La Piola Alba	

Tabella 4: Parametri accettati da /searchByName

### 2.3.3 Esempio di richiesta

```
curl -X POST
https://cs0thtwbr7.execute-api.eu-central-1.amazonaws.com/dev/
searchByName
-H 'Content-Type: application/json'
-d '{"name": "Piazza"}'
```

Listing 5: Esempio di richiesta /searchByName

### 2.3.4 Esempio di risposta

```
[
  {
    "id_ristorante": 9,
    "nome_ristorante": "Piazza Duomo",
    "indirizzo": "",
    "telefono": "+39 0173 366167",
    "sito_web": "http://www.piazzaduomoalba.it/",
    "latitudine": "44.7005",
    "longitudine": "8.0360",
    "categoria": "Italian Restaurant",
    "punteggio_emoji": 16.5644,
    "punteggio_foto": 6.74074,
    "punteggio_testo": 51.2037,
    "url_immagine": "<presigned-url>"
  },
  {
    "id_ristorante": 209,
    "nome_ristorante": "Piazza degli Artisti Ciampino",
    "indirizzo": "",
    "telefono": "+393483623939",
    "sito_web": "",
    "latitudine": "41.8019",
    "longitudine": "12.5994",
    "categoria": "Italian Restaurant",
```

```
[
  {
    "punteggio_emoji": 39.45,
    "punteggio_foto": null,
    "punteggio_testo": 29.7,
    "url_immagine": "<presigned-url>"
  },
  {
    "id_ristorante": 900,
    "nome_ristorante": "Podere La Piazza",
    "indirizzo": "",
    "telefono": "0141966267",
    "sito_web": "http://www.poderelapiazza.it",
    "latitudine": "44.7934",
    "longitudine": "8.1636",
    "categoria": "Restaurant",
    "punteggio_emoji": null,
    "punteggio_foto": 56.7,
    "punteggio_testo": 37.4,
    "url_immagine": "<presigned-url>"
  },
  {
    "id_ristorante": 1578,
    "nome_ristorante": "In Piazza Osteria con Pizza",
    "indirizzo": "",
    "telefono": "0173-787346",
    "sito_web": null,
    "latitudine": "44.5829",
    "longitudine": "7.9677",
    "categoria": "Pizza place",
    "punteggio_emoji": 74.6,
    "punteggio_foto": null,
    "punteggio_testo": 50.0,
    "url_immagine": "<presigned-url>"
  }
]
```

Listing 6: Esempio di richiesta /searchByName

## 2.4 GET - /favorites

Rappresenta il gestore della lista dei preferiti per ogni utente. Permette di fare 3 operazioni:

- L'aggiunta di un nuovo ristorante alla lista dei preferiti di uno specifico utente;
- La rimozione un ristorante dalla lista dei preferiti di uno specifico utente;
- La restituzione della lista dei preferiti per un utente specifico.

La risposta contiene rispettivamente:

- Il risultato dell'operazione (`true` o `false`);
- Il risultato dell'operazione (`true` o `false`);
- Una lista contenente tutti gli id e i nomi dei ristoranti preferiti di uno specifico utente.

### 2.4.1 Endpoint

<https://cs0thtwbr7.execute-api.eu-central-1.amazonaws.com/dev/favorites>

### 2.4.2 Parametri

#### 2.4.2.1 Aggiunta preferito

Nome	Opzionale	Descrizione	Esempio	Default
action	No	L'azione da compiere, il valore da inserire è "add"	add	
user	No	L'identificativo dell'utente	4887b0d7-f041-4a92-a458-eab8aa5f31f	
restaurant	No	L'identificativo del ristorante da inserire	8	

Tabella 5: Parametri accettati da /favorites per l'aggiunta di un preferito

#### 2.4.2.2 Rimozione preferito

Nome	Opzionale	Descrizione	Esempio	Default
action	No	L'azione da compiere, il valore da inserire è "remove"	remove	
user	No	L'identificativo dell'utente	4887b0d7-f041-4a92-a458-eab8aa5f31f	
restaurant	No	L'identificativo del ristorante da rimuovere	8	

Tabella 6: Parametri accettati da /favorites per la rimozione di un preferito



### 2.4.2.3 Restituzione lista preferiti

Nome	Opzionale	Descrizione	Esempio	Default
action	No	L'azione da compiere, il valore da inserire è "get"	get	
user	No	L'identificativo dell'utente	4887b0d7-f041-4a92-a458-eab8aa5f311	

Tabella 7: Parametri accettati da `/favorites` per la restituzione della lista dei preferiti

### 2.4.3 Esempio di richiesta

#### 2.4.3.1 Aggiunta preferito

```
curl -X GET
https://cs0thtwbr7.execute-api.eu-central-1.amazonaws.com/dev/favorites
?action=add&user=515d659b-baec-413b-9b37-703498372093&restaurant=100
```

Listing 7: Esempio di richiesta `/favorites` per l'aggiunta di un preferito

#### 2.4.3.2 Rimozione preferito

```
curl -X GET
https://cs0thtwbr7.execute-api.eu-central-1.amazonaws.com/dev/favorites
?action=remove&user=515d659b-baec-413b-9b37-703498372093&restaurant=100
```

Listing 8: Esempio di richiesta `/favorites` per la rimozione di un preferito

#### 2.4.3.3 Restituzione lista preferiti

```
curl -X GET
https://cs0thtwbr7.execute-api.eu-central-1.amazonaws.com/dev/favorites
?action=get&user=515d659b-baec-413b-9b37-703498372093
```

Listing 9: Esempio di richiesta `/favorites` per la restituzione della lista dei preferiti



## 2.4.4 Esempio di risposta

### 2.4.4.1 Aggiunta preferito

```
true
```

Listing 10: Esempio di risposta `/favorites` per l'aggiunta di un preferito

### 2.4.4.2 Rimozione preferito

```
true
```

Listing 11: Esempio di risposta `/favorites` per la rimozione di un preferito

### 2.4.4.3 Restituzione lista preferiti

```
[{
  "id_ristorante": 8,
  "nome_ristorante": "La Piola Alba"
}, {
  "id_ristorante": 43,
  "nome_ristorante": "Osteria Alla Concorrenza"
}, {
  "id_ristorante": 59,
  "nome_ristorante": "Taverna Trilussa"
}, {
  "id_ristorante": 63,
  "nome_ristorante": "Hotel das Cataratas, A Belmond Hotel"
}, {
  "id_ristorante": 64,
  "nome_ristorante": "Kazuo Restaurante"
}, {
  "id_ristorante": 66,
  "nome_ristorante": "Trattoria Fasano"
}, {
  "id_ristorante": 67,
  "nome_ristorante": "Yokohama Sushi Restaurant"
}, {
  "id_ristorante": 72,
  "nome_ristorante": "ORSO - Laboratorio Caff\u00e8"
}, {
  "id_ristorante": 74,
  "nome_ristorante": "Osteria dei Sognatori"
}]
```

Listing 12: Esempio di risposta `/favorites` per la restituzione della lista dei preferiti



## 2.5 GET - /getCities

Restituisce tutte le informazioni relative alle città presenti nel database. Utilizzata per la ricerca per zona.

### 2.5.1 Endpoint

`https://cs0thtwbr7.execute-api.eu-central-1.amazonaws.com/dev/getCities`

### 2.5.2 Parametri

Nessuno.

### 2.5.3 Esempio di richiesta

```
curl -X GET
https://cs0thtwbr7.execute-api.eu-central-1.amazonaws.com/dev/getCities
```

Listing 13: Esempio di richiesta /getCities

### 2.5.4 Esempio di risposta

```
[
  {
    "nome": "Abano Terme",
    "latitudine": "45.3594",
    "longitudine": "11.7887"
  },
  {
    "nome": "Abbadia Cerreto",
    "latitudine": "45.3124",
    "longitudine": "9.5925"
  },
  {
    "nome": "Abbadia Lariana",
    "latitudine": "45.9005",
    "longitudine": "9.3345"
  },
  {
    "nome": "Abbadia San Salvatore",
    "latitudine": "42.8810",
    "longitudine": "11.6775"
  },
  ...
]
```

Listing 14: Esempio di richiesta /getCities





## 2.6 GET - /getLabelCategoria

Restituisce tutte le label<sup>G</sup> presenti nel database relative alle categorie di cucina dei ristoranti.

### 2.6.1 Endpoint

`https://cs0thtwbr7.execute-api.eu-central-1.amazonaws.com/dev/getLabelCategoria`

### 2.6.2 Parametri

Nessuno.

### 2.6.3 Esempio di richiesta

```
curl -X GET
https://cs0thtwbr7.execute-api.eu-central-1.amazonaws.com/dev/
getLabelCategoria
```

Listing 15: Esempio di richiesta /getLabelCategoria

### 2.6.4 Esempio di risposta

```
[
  {
    "categorie": "Italian Restaurant"
  },
  {
    "categorie": "Restaurant"
  },
  {
    "categorie": "Piedmont Restaurant"
  },
  {
    "categorie": "Diner"
  },
  {
    "categorie": "Wine/spirits"
  },
  {
    "categorie": "Chinese Restaurant"
  },
  {
    "categorie": "Cocktail Bar"
  },
  ...
]
```

Listing 16: Esempio di richiesta /getLabelCategoria

## 2.7 GET - /generalFilter

Filtra la classifica per località (tramite un raggio in km) e per categoria di cucina, e per ogni ristorante ritorna le informazioni definite in §2.1.

### 2.7.1 Endpoint

<https://cs0thtwbr7.execute-api.eu-central-1.amazonaws.com/dev/generalFilter>

### 2.7.2 Parametri

Nome	Opzionale	Descrizione	Esempio	Default
radius	Sì	Raggio in km dalla location	8	10
location	Sì	Il nome del ristorante da cercare	Milano	
categoria	Sì	Il nome del ristorante da cercare	Hotel	
position	Sì	Il nome del ristorante da cercare	0	0
size	Sì	Il nome del ristorante da cercare	10	10

Tabella 8: Parametri accettati da /generalFilter

Il parametro "radius" viene considerato solo se anche il parametro "location" è specificato.

### 2.7.3 Esempio di richiesta

```
curl -X GET
https://cs0thtwbr7.execute-api.eu-central-1.amazonaws.com/dev/
  getLabelCategoria?radius=10&location=Milano&categoria=Hotel&position
  =0&size=10
```

Listing 17: Esempio di richiesta /searchByName

### 2.7.4 Esempio di risposta

```
[
  {
    "id_ristorante": 2839,
    "nome_ristorante": "The Westin Palace, Milan",
    "indirizzo": "",
    "telefono": "+390263361",
    "sito_web": "https://www.marriott.com/hotels/travel/milwi-the-
      westin-palace-milan/?scid=5d6c9ba8-fdb3-45cd-90d4-
      fc3cbfe3681c&y_source=1
      _Mjc4MjEzMjE0NTktbG9jYXRpb24ud2Vic2l0ZQ%3D%3D",
    "latitudine": "45.4786",
    "longitudine": "9.1992",
    "categoria": "Hotel",
    "punteggio_emoji": 79.8875,
    "punteggio_foto": 100.0,
    "punteggio_testo": 70.9667,
    "url_image": "<presigned-url>"
  },
  {
    "id_ristorante": 2156,
    "nome_ristorante": "Four Seasons Hotel Milan",
```

```
[
  {
    "indirizzo": "",
    "telefono": "+390277088",
    "sito_web": "http://www.fourseasons.com/",
    "latitudine": "45.4693",
    "longitudine": "9.1955",
    "categoria": "Hotel",
    "punteggio_emoji": 86.375,
    "punteggio_foto": null,
    "punteggio_testo": 72.4,
    "url_image": "<presigned-url>"
  },
  {
    "id_ristorante": 550,
    "nome_ristorante": "Principe di Savoia",
    "indirizzo": "",
    "telefono": "+390262301",
    "sito_web": "https://www.dorchestercollection.com/en/milan/hotel-principe-di-savoia",
    "latitudine": "45.4799",
    "longitudine": "9.1965",
    "categoria": "Hotel",
    "punteggio_emoji": 84.325,
    "punteggio_foto": 82.5,
    "punteggio_testo": 62.375,
    "url_image": "<presigned-url>"
  },
  ...
]
```

Listing 18: Esempio di richiesta /getLabelCategoria



## 3 API Crawling Service

### 3.1 POST - /test\_post

Permette di suggerire un profilo che verrà poi utilizzato dal crawler per estrarre i dati. Verranno solo analizzati i profili pubblici. L'operazione può ritornare quattro risultati, identificati da un numero intero:

- 0 (SUCCESS): profilo è stato aggiunto con successo;
- 1 (ALREADY\_IN\_DB): Il profilo è già nel database;
- 2 (DOES\_NOT\_EXIST): Il profilo non esiste;
- 3 (PRIVATE\_PROFILE): Il profilo è privato.

#### 3.1.1 Endpoint

`https://eultdbw3g5.execute-api.eu-central-1.amazonaws.com/dev/test_post`

#### 3.1.2 Parametri

Nome	Opzionale	Descrizione	Esempio	Default
name	No	Il nome utente del profilo da aggiungere	francesco	

Tabella 9: Parametri accettati da /test\_post

#### 3.1.3 Esempio di richiesta

```
curl -X POST
https://eultdbw3g5.execute-api.eu-central-1.amazonaws.com/dev/test_post
-H 'Content-Type: application/json'
-d '{"name": "sfpugnfnsgnfp"}'
```

Listing 19: Esempio di richiesta /test\_post

#### 3.1.4 Esempio di risposta

```
{
  "message": "Profile does not exist",
  "status": 2
}
```

Listing 20: Esempio di risposta /test\_post