# Piattaforma: Twitter

## Documentazione della piattaforma

L’obiettivo di questa sezione è facilitare la comprensione dei dati prodotti dalla piattaforma mediante l’analisi della documentazione rilasciata dalla piattaforma stessa.

Twitter mette a disposizione tre prodotti – products – per poter interagire con le funzionalità ed i dati messi a disposizione dalla piattaforma:

* Twitter API
* Twitter Ads API
* Twitter for Websites

Per sviluppare un progetto di social media mining ed avere accesso ai dati “social” forniti da Twitter si utilizza il prodotto Twitter API.

### Link alla documentazione

Il punto di accesso principale alla documentazione delle Twitter API è <https://developer.twitter.com/en/docs/twitter-api>. Attualmente la piattaforma raccomanda di utilizzare la nuova versione delle Twitter API: la versione v2.0.

La documentazione presenta una serie di link alle risorse fondamentali ed ai concetti fondamentali per l’utilizzo efficace delle API:

|  |  |
| --- | --- |
| Risorsa | URL |
| Developers Apps | https://developer.twitter.com/en/docs/apps/overview |
| Project | https://developer.twitter.com/en/docs/projects |
| Authentication | https://developer.twitter.com/en/docs/authentication |
| Rate Limit | https://developer.twitter.com/en/docs/rate-limits |
| Twitter ID | https://developer.twitter.com/en/docs/twitter-ids |

Una descrizione sintetica di tutte le risorse implementate dalle API ed i relativi URL sono disponibili all’indirizzo <https://developer.twitter.com/en/docs/api-reference-index> . In generale la piattaforma ha adottato il classico modello di API di tipo RESTFull. Il protocollo a livello applicativo per inoltrare le richieste è quindi HTTP.

### Individuare i dati disponibili

Mediante gli endpoint di Twitter API possono accedere ad una serie di entità, chiamate object. Ogni endpoint restituisce una risorsa di primo livello che corrisponde ad uno degli object implementati dalla piattaforma: Tweet, User, Space, List, Media, Poll, Place

#### Profilo account

Un solo tipo di Twitter object è riconducibile al concetto di profilo: l’object User. Tale oggetto contiene tutti i metadati che descrivono un account Twitter. Nello specifico i meta-dati associati ad un User object sono i seguenti (tabella adattata presa dalla documentazione - ndA):

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Descrizione** |
| id | string | Identificatore univoco sull’insieme di tutti i Twitter Object. E’ un’informazione **immutabile**, rimane lo stesso per tutto il periodo di vita dello User object. |
| name | string | Nome dell’utente, non necessariamente corrispondente al nome e cognome dell’utente |
| username (default) | string | Detto anche screen name. Indica il nickname scelto dall’utente per identifcarsi. Identificatore univoco, ma **può cambiare** durante il periodo di vita di un account. |
| created\_at | date (ISO 8601) | Data di creazione dell’account, nel formato YYYY-MM-DD*T*hh:mm:ss.msZ . Il fuso orario è UTC – Greenwich. La data di creazione viene assegnata dalla piattaforma. |
| description | string | Testo di presentazione dell’account. Il testo viene inserito dall’utente ed è quindi un dato piuttosto rumoroso, non essendo vincolato o generato dalla piattaforma. |
| entities | object | Oggetto complesso in cui sono specificate le entità rilevanti identificate dalla piattaforma quali url, hashtag, menzioni. I meta-dati associati ad ogni entità saranno oggetto in una specifica sezione del documento. Di seguito un esempio:  "entities": {        "url": {            "urls": [                {                    "start": 0,                    "end": 23,                    "url": "https://t.co/3ZX3TNiZCY",                    "expanded\_url": "/content/developer-twitter/en/community",                    "display\_url": "developer.twitter.com/en/community"                }            ]        },        "description": {            "urls": [                {                    "start": 0,                    "end": 23,                    "url": "https://t.co/3ZX3TNiZCY",                    "expanded\_url": "/content/developer-twitter/en/community",                    "display\_url": "developer.twitter.com/en/community"                }]           }    } |
| location | string | Posizione indicata dall’utente. Dato molto rumoroso in quanto viene compilato dall’utente. Molto spesso non indica una posizione valida. |
| pinned\_tweet\_id | string | Identificatore univoco del tweet fissato. Un tweet fissato è un post che viene sempre visualizzato all’inizio della timeline dell’utente a prescindere dalla data di creazione del tweet stesso. Il valore può cambiare a seconda delle scelte dello User. |
| profile\_image\_url | string | URL dell’immagine del profilo di User |
| protected | boolean | Indica se il tweet dell’utente sono privati. Limita l’accesso ai dati forniti dalla piattaforma. |
| public\_metrics | object | Informazioni circa il numero di follower, following, tweet prodotti, ed in quante liste compare. |
| url | string | URL indicato dall’utente. Può non essere definito. |
| verified | boolean | Indica se l’account è verificato |
| withheld | object | Dettagli sui contenuti non visualizzabili (marginale per molti progetti) |

#### Relazioni tra account

Le Twitter API prevedono un solo object che esplicita modella le relazioni tra User object: List object. Un List object è una lista di utenti a cui viene associato un nome e una descrizione. La lista viene creata da un User object e può essere seguita da altri User object. La lista può essere privata o pubblica. La lista è gestita dall’utente che l’ha creata ma un membro della lista può richiedere la rimozione del suo account dalla lista. Nel caso la lista sia pubblica, è possibile conoscere i membri della lista. **Non è chiaro quanto sia diffuso l’utilizzo delle liste per la gestione delle relazioni tra account in Twitter**.

Oltre alla precedente object, Twitter mette a disposizione la funzionalità di “follow” che permette di creare una relazione asimmetrica esplicita tra un account che “segue” – follower – e un account che viene seguito – friend del follower (gergo di Twitter). Le Twitter API non rilasciano una risorsa che modella la relazione di follow, tuttavia mettono a disposizione una risorsa/URL che permette di ottenere la lista dei folllower e dei friends di un account. Tale lista contiene solo gli ID degli account, è ordinata in ordine cronologico – dalla relazione più recente alla meno recente, e viene applicato il meccanismo di paginazione nel caso la lunghezza della lista ecceda un valore di soglia.

Oltre alla relazione di “follow” esiste delle relazioni tra account che sono mediate dai contenuti creabili in Twitter (vedi sezione successiva): mentions, retweet, quote, reply/conversation. La **mention** è un riferimento ad un User object A in un tweet prodotto dallo user B. In questo caso avremo una relazione da B verso A. La condivisione di un tweet o **retweet**, prevede che un utente B condivida un tweet creato da A. In questo caso avremo una relazione da B verso A. Infine, un tweet creato da A può essere “**quotato”** – incluso – in un tweet creato dall’utente B. In questo caso si crea una relazione B verso A. E’ possibile, inoltre, creare, da parte di un utente B, un tweet in risposta ad un tweet creato da B, e a sua volta A può rispondere al tweet creato da B oppure un utente C può rispondere ai tweet precedentemente creati. In questo caso le nuove API associano alla conversazione tra A, B e C un identificatore – conversation id – che corrisponde al tweet id che ha originato la conversazione. In questo caso si definisce un relazione da chi risponde verso l’utente che ha ricevuto il commento.

#### Contenuti accessibili e meta-dati dei contenuti

Dal momento che Twitter è principalmente una piattaforma di micro-blogging, il contenuto principale è il tweet. Un tweet viene modellato dall’oggetto **Tweet**. L’oggetto Tweet è piuttosto complesso in quanto contiene molte proprietà che si riferiscono ad altri oggetti delle API.

Di seguito viene riportata una tabella riassuntiva (adattamento della tabella presente nella documentazione)

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Descrizione** |
| id (default) | string | Identificatore univoco di un tweet. Anche un retweet viene considerato un tweet nuovo. Informazione immutabile. |
| text (default) | string | Testo codificato mediante UTF-8 associato al tweet. Molte delle proprietà associate al tweet dipendono dal testo opportunamente processato dalla piattaforma per estrarne entità significative. |
| attachments | object | Allegati al tweet  "attachments": {     "poll\_ids": [         "1199786642468413448"     ] }  "attachments": {     "media\_keys": [         "3\_1136048009270239232"     ] } |
| author\_id | string | Id dello User object autore del tweet. Dato immutabile |
| context\_annotations | array | Annotazioni estratte in modo automatico da Twitter dal testo del tweet. La qualità delle annotazioni dipende dall’algoritmo di entity recognition implementato dalla piattaforma  "context\_annotations": [       {            "domain": {                "id": "45",                "name": "Brand Vertical",                "description": "Top level entities that describe a Brands industry"            }        },        {            "domain": {                "id": "46",                "name": "Brand Category",                "description": "Categories within Brand Verticals that narrow down the scope of Brands"            },            "entity": {                "id": "781974596752842752",                "name": "Services"            }        },        {            "domain": {                "id": "47",                "name": "Brand",                "description": "Brands and Companies"            },            "entity": {                "id": "10045225402",                "name": "Twitter"            }        }    ] |
| conversation\_id | string | Tweet ID che ha iniziato la conversazione. Vedi spiegazione nella sezione precedente |
| created\_at | date (ISO 8601) | Data di creazione del tweet. Il formato della data è lo stesso visto per User object. La data viene fornita dalla piattaforma ed è immutabile |
| entities | object | Entità estratte dalla piattaforma: mentions, hashtags, cashtags ($apple), urls. Ogni entità è un JSON Object -> dict in Python J. La qualità dell’estrazione dipende dalla piattaforma.  "entities": {     "annotations": [         {            "start": 144,            "end": 150,            "probability": 0.626,            "type": "Product",            "normalized\_text": "Twitter"         }     ],    "cashtags": [         {             "start": 18,             "end": 23,             "tag": "twtr"         }     ],     "hashtags": [         {             "start": 0,             "end": 17,             "tag": "blacklivesmatter"         }     ],     "mentions": [         {             "start": 24,             "end": 35,             "tag": "TwitterDev"         }     ],     "urls": [         {            "start": 44,            "end": 67,            "url": "https://t.co/crkYRdjUB0",            "expanded\_url": "https://twitter.com",            "display\_url": "twitter.com",            "status": "200",            "title": "bird",            "description": "From breaking news and entertainment to sports and politics, get the full story with all the live commentary.",             "unwound\_url": "https://twitter.com"         }     ]} |
| geo | object | JSON Object contenente I dettagli della posizione inserita dall’utente nel tweet  "geo": {        "coordinates": {            "type": "Point",            "coordinates": [                -73.99960455,                40.74168819            ]        },        "place\_id": "01a9a39529b27f36"    } |
| in\_reply\_to\_user\_id | string | ID dello User object che ha creato il tweet a cui l’oggetto Tweet risponde |
| lang | string | Lingua utilizzata per scrivere il tweet. Identificata in modo automatico dalla piattaforma |
| possibly\_sensitive | boolean | Indica se il contenuto referenziato dall’URL presente nel tweet – qualora esista – è “sensibile”. Il giudizio viene emesso dalla piattaforma |
| public\_metrics | object | Metriche di engagement pubblicamente disponibili. Per la natura della piattaforma, le metriche saturano circa mezza giornata dopo la pubblicazione. Dal momento della creazione ci si aspetta un primo periodo di accelerazione seguito incrementi più sporadici. Aspetti dinamici da prendere in considerazione.  "public\_metrics" : {          "retweet\_count": 8,          "reply\_count": 2,          "like\_count": 39,          "quote\_count": 1  } |
| referenced\_tweets | array | Lista di tweet a cui questo Tweet si riferisce  "referenced\_tweets": [        {            "type": "replied\_to",            "id": "1242125486844604425"        }    ]  "referenced\_tweets": [        {            "type": "quoted",            "id": "1265712391578214400"        }    ] |
| reply\_settings | string | Mostra chi può rispondere al tweet: "everyone", "mentioned\_users", o "followers". Scelta espressa dall’utente |
| source | string | Applicazione utilizzato per creare il tweet. |

Per l’identificazione delle interazioni e relazioni implicite tra utenti si possono utilizzare i valori associati alle proprietà: *referenced\_tweets*, *in\_reply\_to\_user\_id*, *entities.mentions*.

## Estrazione dei dati

### Protocollo di accesso e autorizzazione all’accesso ai dati

Twitter API implementa il flusso di autorizzazione e autenticazione di OAuth2 detto client credential grant. E’ necessario quindi creare un’applicazione Twitter – Developer App – per ottenere una API Key e una API Secret che rappresentano la credenziali di accesso all’applicazione. Per creare un’applicazione si deve registrare un account developer e seguire il processo di registrazione al seguente link <https://developer.twitter.com/en/portal/petition/essential/basic-info>. Una volta completato il processo di può accedere alla dashboard degli sviluppatori – Developer Portal. In Projects & Apps sono visibili le applicazioni attivate. Selezionando un’applicazione si ha accesso ai dettagli. Gli chiavi ed i token di accesso sono riportate nel tab “Keys and tokens”.

Queste informazioni sono necessarie per implementare il flusso di client credential. Questo flusso viene chiamato in Twitter OAuth2.0 Bearer Token ed è descritto in dettaglio al seguente link https://developer.twitter.com/en/docs/authentication/oauth-2-0

### Endpoint di accesso ai dati identificati

La documentazione riporta una descrizione riepilogativa di tutte le risorse e i link alla relativa documentazione approfondita. In seguito pubblichiamo una tabella che si riferisce solo ai dati e alle entità descritte in precedenza nel documento:

## Tweets

### COVID-19 stream

* + [GET /labs/1/tweets/stream/compliance](https://developer.twitter.com/en/docs/twitter-api/tweets/covid-19-stream/api-reference/get-tweets-stream-compliance)
  + [GET /labs/1/tweets/stream/covid19](https://developer.twitter.com/en/docs/twitter-api/tweets/covid-19-stream/api-reference/get-tweets-stream-covid19)

### Filtered stream

* + [GET /2/tweets/search/stream](https://developer.twitter.com/en/docs/twitter-api/tweets/filtered-stream/api-reference/get-tweets-search-stream)
  + [GET /2/tweets/search/stream/rules](https://developer.twitter.com/en/docs/twitter-api/tweets/filtered-stream/api-reference/get-tweets-search-stream-rules)
  + [POST /2/tweets/search/stream/rules](https://developer.twitter.com/en/docs/twitter-api/tweets/filtered-stream/api-reference/post-tweets-search-stream-rules)

### Likes

* + [GET /2/tweets/:id/liking\_users](https://developer.twitter.com/en/docs/twitter-api/tweets/likes/api-reference/get-tweets-id-liking_users)
  + [GET /2/users/:id/liked\_tweets](https://developer.twitter.com/en/docs/twitter-api/tweets/likes/api-reference/get-users-id-liked_tweets)

### Quote Tweets

* + [GET /2/tweets/:id/quote\_tweets](https://developer.twitter.com/en/docs/twitter-api/tweets/quote-tweets/api-reference/get-tweets-id-quote_tweets)

### Retweets

* + [GET /2/tweets/:id/retweeted\_by](https://developer.twitter.com/en/docs/twitter-api/tweets/retweets/api-reference/get-tweets-id-retweeted_by)

### Search Tweets

* + [GET /2/tweets/search/recent](https://developer.twitter.com/en/docs/twitter-api/tweets/search/api-reference/get-tweets-search-recent)

### Timelines

* + [GET /2/users/:id/mentions](https://developer.twitter.com/en/docs/twitter-api/tweets/timelines/api-reference/get-users-id-mentions)
  + [GET /2/users/:id/tweets](https://developer.twitter.com/en/docs/twitter-api/tweets/timelines/api-reference/get-users-id-tweets)

### Tweet counts

* + [GET /2/tweets/counts/all](https://developer.twitter.com/en/docs/twitter-api/tweets/counts/api-reference/get-tweets-counts-all)
  + [GET /2/tweets/counts/recent](https://developer.twitter.com/en/docs/twitter-api/tweets/counts/api-reference/get-tweets-counts-recent)

### Tweets lookup

* + [GET /2/tweets](https://developer.twitter.com/en/docs/twitter-api/tweets/lookup/api-reference/get-tweets)
  + [GET /2/tweets/:id](https://developer.twitter.com/en/docs/twitter-api/tweets/lookup/api-reference/get-tweets-id)

### Volume streams

* + [GET /2/tweets/sample/stream](https://developer.twitter.com/en/docs/twitter-api/tweets/volume-streams/api-reference/get-tweets-sample-stream)

## Users

### Blocks

* + [GET /2/users/:id/blocking](https://developer.twitter.com/en/docs/twitter-api/users/blocks/api-reference/get-users-blocking)

### Follows

* + [GET /2/users/:id/followers](https://developer.twitter.com/en/docs/twitter-api/users/follows/api-reference/get-users-id-followers)
  + [GET /2/users/:id/following](https://developer.twitter.com/en/docs/twitter-api/users/follows/api-reference/get-users-id-following)

### Mutes

* + [GET /2/users/:id/muting](https://developer.twitter.com/en/docs/twitter-api/users/mutes/api-reference/get-users-muting)

### Users lookup

* + [GET /2/users](https://developer.twitter.com/en/docs/twitter-api/users/lookup/api-reference/get-users)
  + [GET /2/users/:id](https://developer.twitter.com/en/docs/twitter-api/users/lookup/api-reference/get-users-id)
  + [GET /2/users/by](https://developer.twitter.com/en/docs/twitter-api/users/lookup/api-reference/get-users-by)
  + [GET /2/users/by/username/:username](https://developer.twitter.com/en/docs/twitter-api/users/lookup/api-reference/get-users-by-username-username)

## Lists

### List Tweets lookup

* + [GET /2/lists/:id/tweets](https://developer.twitter.com/en/docs/twitter-api/lists/list-tweets/api-reference/get-lists-id-tweets)

### List follows

* + [GET /2/lists/:id/followers](https://developer.twitter.com/en/docs/twitter-api/lists/list-follows/api-reference/get-lists-id-followers)
  + [GET /2/users/:id/followed\_lists](https://developer.twitter.com/en/docs/twitter-api/lists/list-follows/api-reference/get-users-id-followed_lists)

### List lookup

* + [GET /2/lists/:id](https://developer.twitter.com/en/docs/twitter-api/lists/list-lookup/api-reference/get-lists-id)
  + [GET /2/users/:id/owned\_lists](https://developer.twitter.com/en/docs/twitter-api/lists/list-lookup/api-reference/get-users-id-owned_lists)

### List members

* + [GET /2/lists/:id/members](https://developer.twitter.com/en/docs/twitter-api/lists/list-members/api-reference/get-lists-id-members)
  + [GET /2/users/:id/list\_memberships](https://developer.twitter.com/en/docs/twitter-api/lists/list-members/api-reference/get-users-id-list_memberships)

Gli endpoint relativi all’oggetto Tweet mettono a disposizione un meccanismo di ricerca i cui risultati variano in base al tipo di account developer: la documentazione dettagliata sugli operatori di ricerca è disponibile al seguente indirizzo: <https://developer.twitter.com/en/docs/twitter-api/tweets/search/integrate/build-a-query>

## Software

### Librerie per la piattaforma

La documentazione contiene riferimenti ai tools e alle librerie sviluppate da Twitter o da terze parti per la maggior parte dei linguaggi di programmazione. Per questo progetto scegliamo la libreria tweepy in quanto si interfaccia con tutte le versioni delle Twitter API. La documentazione della libreria è disponibile al seguente indirizzo: https://docs.tweepy.org/en/latest/index.html