1506
UNIVERSITÀ
DEGLI STUDI
DI URBINO
CARLO BO

CORSO DI LAUREA IN
INFORMATICA APPLICATA
SCUOLA DI
SCIENZE TECNOLOGIE E FILOSOFIA DELL'INFORMAZIONE

Reti di Calcolatori – Progetto a.a. 2020/2021

"Scraping Instagram"

Sviluppatori:

Simone Cossi, mat. 290796

INDICE

INTRODUZIONE

L’obiettivo principale del progetto è quello di approfondire la mia conoscenza riguardante una metodologia di raccolta delle informazioni comunemente denominata **scraping**. Lo scraping è una tecnica informatica di estrazione di dati da un sito web per mezzo di programmi software. Di solito, tali programmi simulano la navigazione umana nel World Wide Web utilizzando l'Hypertext Transfer Protocol (HTTP) o attraverso browser, come Internet Explorer o Mozilla Firefox.

In particolare, il mio progetto andrà a concentrarsi sullo scraping di dati ed informazioni relativi ad Instagram, noto e diffuso social, per poter andare a cercare informazioni che sono state nascoste dagli sviluppatori stessi in seguito a diversi aggiornamenti. Tra le informazioni troviamo:

- Numero di like
- Screenshot
- Etc.

Il progetto in questione è stato sviluppato per poter funzionare da terminale; lo sviluppo di una GUI, per quanto utile, sembrava superfluo.

Specifica del Progetto

Programma che permette a linea di comando di inviare e ricevere dati da siti web tramite http, una sorta di console del browser.

Realizzazione del progetto

Nell’atto pratico ciò che il programma va ad eseguire sono due diverse richieste http.

Le richieste vengono effettuate a due API diverse, in modo da ottenere contenuti di diverso tipo:

1) Nel primo caso verranno restituiti tutti i dati riguardanti un profilo Instagram di cui ne verranno letti solo alcuni e stampati a video su console.

2) Nel secondo caso verranno restituiti tutti i dati riguardanti i post di Instagram che avranno utilizzato lo stesso #hashtag, questi dati verranno letti e filtrati e alcuni di questi verranno inseriti in un file .html

SCELTE PROGETTUALI

Ho scelto innanzitutto di utilizzare delle API di terzi poiché dopo svariati tentativi senza successo di leggere le informazioni direttamente del html, mi sono imbattuto sul consiglio di provare ad utilizzare delle API, raggiungendo così il mio obbiettivo.

Il funzionamento di entrambe le API è abbastanza semplice:

Si effettuano delle richieste http personalizzandole come indicato degli sviluppatori, viene restituito un file json contenente le informazioni.

Definizione di API:

[2] Le API (acronimo di Application Programming Interface, ovvero Interfaccia di programmazione delle applicazioni) sono set di definizioni e protocolli con i quali vengono realizzati e integrati software applicativi. Consentono ai tuoi prodotti o servizi di comunicare con altri prodotti o servizi senza sapere come vengono implementati, semplificando così lo sviluppo delle app e consentendo un netto risparmio di tempo e denaro.

Linguaggio di programmazione

Per lo sviluppo dello script è stato scelto il Python.

La scelta è dovuta al grande numero di librerie fornite dalla community per svolgere i compiti più diversi e al fatto che Python, essendo un linguaggio interpretato, necessita solamente del suo interprete per poter funzionare su qualsiasi dispositivo (perché supporti l'interprete);

Recupero dei dati

Per recuperare i dati si è deciso di fare uso del protocollo HTTP: nello specifico il client (il tool sviluppato), ogni qualvolta deve interagire con il server, invia delle richieste. Di conseguenza, il server, ad ogni richiesta manda delle risposte.

Generalmente le risposte con cui mi sono imbattuto erano in formato html o in formato json, per comodità di sviluppo il programma finale utilizza solo risposte in formato json.

Lettura delle informazioni

Come è stato appena detto il programma lavora principalmente con risposte in codice json.

Fondamentale per andare a leggere e utilizzare la risposta in json è stata una libreria python chiamata ‘json’[1] che mi permette di convertire il json in un oggetto python. Una volta convertito è facilmente leggibile, perciò si procede con l’acquisizione delle informazioni desiderate.

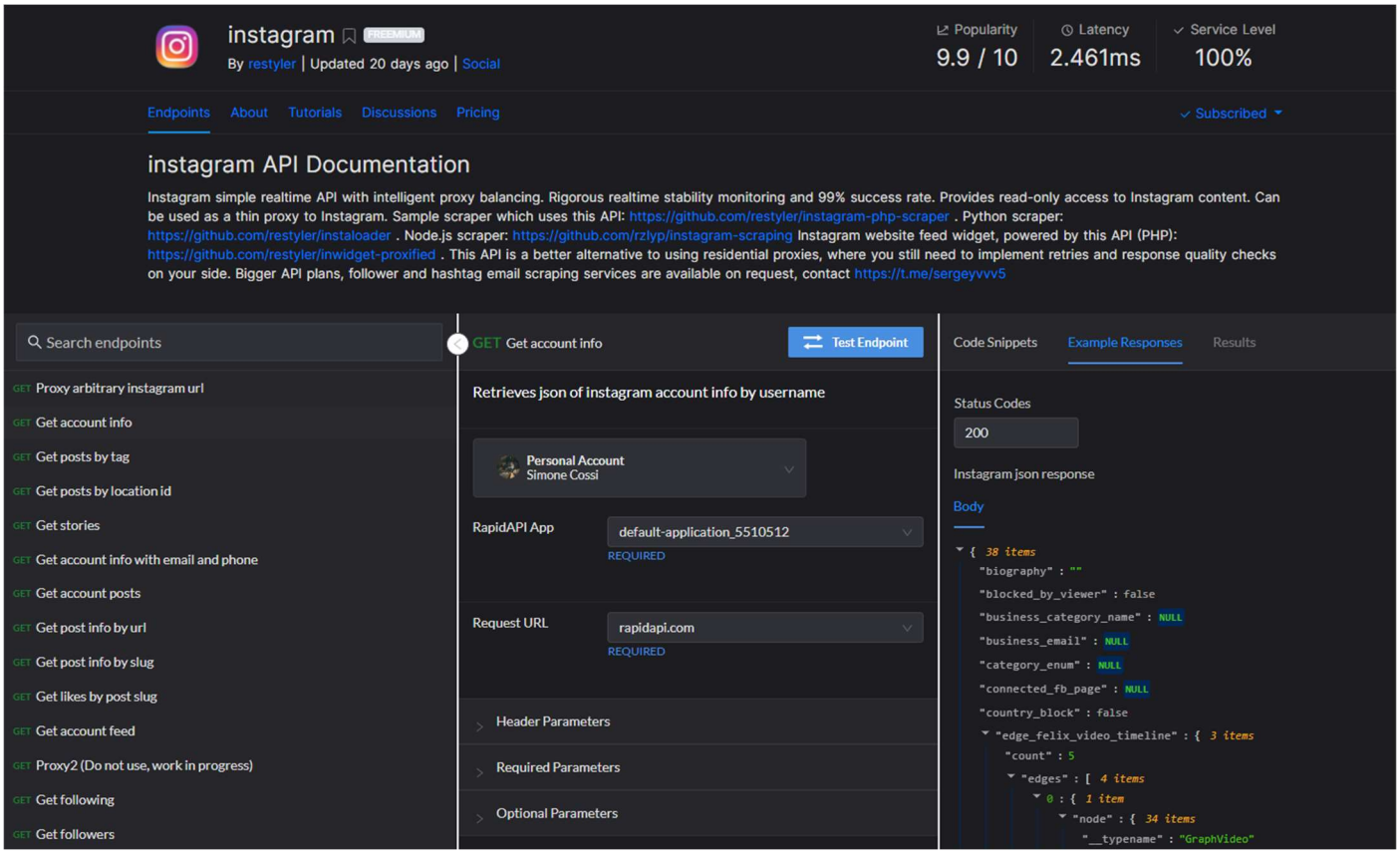
STRUTTURA DELLE RISPOSTE

Sono state utilizzate due API differenti di due sviluppatori differenti poiché essendo sviluppate da terzi, questi lasciano un utilizzo minimo di prova gratuito in un determinato lasso di tempo; perciò, dopo aver esaurito gli utilizzi di diverse API e non potendole più utilizzare per il resto del mese (avrebbe causato l'impossibilità di test e prova da parte del professore) ho dovuto optare per API diverse. Questo è lo stesso motivo per cui il programma non contiene molte altre funzioni, con l'aggiunta di queste sarebbero aumentati i test e diminuiti drasticamente gli utilizzi e avrebbero potuto causare la rimozione di altre funzioni.

Info account

Effettuare una richiesta http che ritorni le informazioni di un account desiderato tramite l'API 'instagram' è semplice.

Per ottenere le informazioni tramite questa API basta effettuare una connessione all'URL indicato e



proseguire con una richiesta 'GET', strutturandola come indicato dagli sviluppatori e inserendo il nome utente desiderato nella posizione corretta. Per ogni chiarimento sulla risposta lascio un collegamento diretto al sito dell'API: instragram[3]

Guida per l'endpoint utilizzato:

Nella prima colonna selezionare: **Get account info**

Nella terza colonna selezionare in alto:

- Code Snippets e selezionare il linguaggio per avere l'implementazione nel codice nel linguaggio che si preferisce
- Example Responses per avere un esempio della struttura della risposta. Grazie a ciò è stato semplificato lo sviluppo delle istruzioni di lettura e acquisizione delle sole informazioni desiderate.

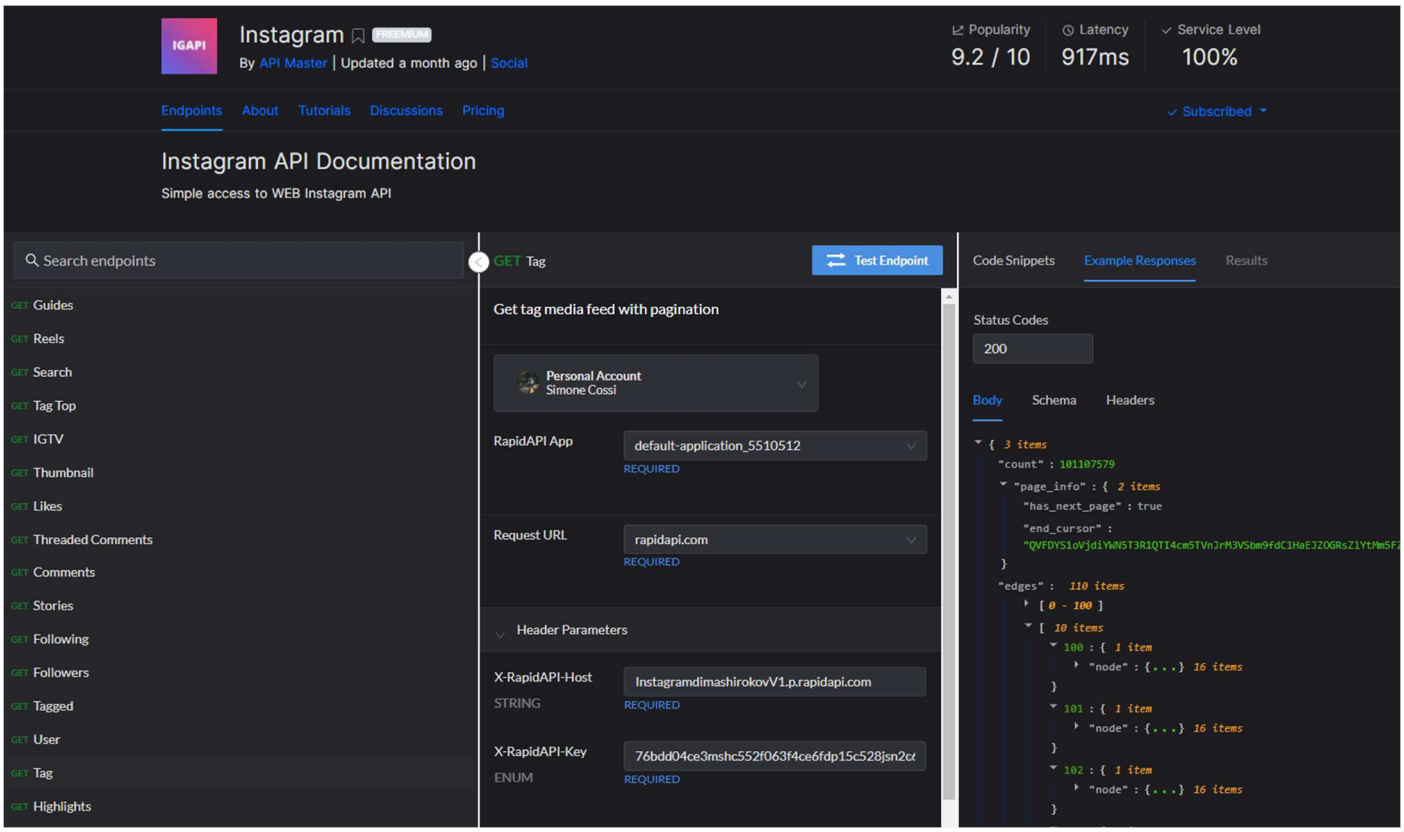
Tag Post

Per l'acquisizione, invece, dei post con uno stesso comune tag (hashtag) è stato necessario l'utilizzo di una diversa API. In questo caso l'API si chiama 'Instagram' (sembra uguale ma cambia la maiuscola iniziale).

L'acquisizione dei dati è stata semplice come nel caso precedente, cambiano un paio di parametri e la struttura della richiesta, ma il funzionamento è pressoché identico.

Le complicazioni sono arrivate con la lettura e l'acquisizione delle informazioni. Precedentemente era bastato leggere l'oggetto python ottenuto dalla conversione del json nei punti giusti, in questo caso l'oggetto è più complesso e le informazioni desiderate sono maggiori; perciò, si è optato per un ciclo analizza singolarmente le varie parti dell'oggetto.

Per ogni chiarimento sulla risposta lascio un collegamento diretto al sito dell'API: [Instagram](#)[4]



Guida per l'endpoint utilizzato:

Nella prima colonna selezionare: **Tag**

Nella terza colonna selezionare in alto:

- Code Snippets e selezionare il linguaggio per avere l'implementazione nel codice nel linguaggio che si preferisce
- Example Responses per avere un esempio della struttura della risposta. Grazie a ciò è stato semplificato lo sviluppo delle istruzioni di lettura e acquisizione delle sole informazioni desiderate.

PROGRAMMA PRINCIPALE

Il programma principale è una semplice console, all'inizio verrà richiesto che tipo di ricerca effettuare, successivamente in base alla prima scelta ci saranno due percorsi diversi:

Primo caso – info account

Nel primo caso verrà richiesto un nome utente, verrà effettuata la ricerca delle informazioni riguardanti l'account relativo al nome utente inserito e verranno stampate a video, sempre su console, alcune informazioni:

- Nome completo
- Bio
- Numero di Followers
- Numero di persone seguite
- Link per la visualizzazione dell'immagine del profilo

Secondo caso – tag post

Nel secondo caso verrà, invece, richiesto un 'hashtag', il programma cercherà le foto che utilizzano lo stesso hashtag indicato e acquisendone alcune informazioni creerà un file .html che permetterà di visualizzare alcune informazioni e la foto.

- Numero di like
- Descrizione

CONCLUSIONE

Lo scraping dei dati è andato a buon fine nonostante un paio di intoppi e altrettanti limitazioni date soprattutto dall'inesperienza in questo ambito e dalle limitazioni di tempo.

Confronto con l'obbiettivo iniziale

L'obbiettivo iniziale purtroppo è stato raggiunto solo in parte, le conoscenze e le competenze nell'ambito sono decisamente migliorate, nonostante ciò, non è stato raggiunto l'obbiettivo di una console che mostrasse tutte le informazioni desiderate, soprattutto quelle normalmente nascoste agli utenti comuni.

Sebbene non sia stata sviluppata una console con tutte le funzionalità desiderate inizialmente sono state raggiunte le conoscenze per attuare il tutto.

Il fatto che il programma non le renda pubbliche è solo un discorso di preferenze e cambio di idea durante lo sviluppo del codice. Al posto di mostrare contenuti 'sensibili' durante l'esame si è preferito mostrare contenuti 'pubblici' dimostrando però di aver raggiunto le competenze desiderate.

Concludendo:

Ci sono nuovi obbiettivi nello stesso ambito?

La risposta è sicuramente affermativa, innanzitutto, il prossimo step sarà sicuramente imparare ad effettuare uno scraping dei dati senza l'utilizzo e l'aiuto di API sviluppate da terzi.

SITOGRAFIA

1) https://docs.python.org/3/library/json.html#
   Documentazione sulla libreria ‘json’ utilizzata

2) https://www.redhat.com/it/topics/api/what-are-application-programming-interfaces

3) https://rapidapi.com/restyler/api/instagram40/

4) https://rapidapi.com/v.sobolev/api/Instagram/

5) https://docs.python.org/3/library/http.client.html?highlight=http%20client#
   Documentazione sulla libreria ‘http.client’ utilizzata per eseguire richieste http