



(<http://codingjam.it/>)

f (<https://www.facebook.com/cosenonjaviste>) **t** (<https://twitter.com/codingjam>) **g+**

(<https://plus.google.com/u/0/+cosenonjavisteit>) **in**

([https://www.linkedin.com/company/5368413?](https://www.linkedin.com/company/5368413?trk=tyah&trkInfo=clickedVerticalcompanyclickedEntityId5368413idx1-1-1tarId1458936352847tascosenon)

[1tarId1458936352847tascosenon](https://www.linkedin.com/company/5368413?trk=tyah&trkInfo=clickedVerticalcompanyclickedEntityId5368413idx1-1-1tarId1458936352847tascosenon))

YouTube

(<https://www.youtube.com/channel/UCMNLvAHV2vIMbD5hWIsUjFA>) **rss**

(<http://codingjam.it/feed/>)



Una introduzione ai bot di Telegram (<http://codingjam.it/introduzione-bot-telegram/>)

28. Oct. 2015 / Tutorial (<http://codingjam.it/tutorial-2/>) /

3 Comments (http://codingjam.it/introduzione-bot-telegram/#disqus_thread)

Lo scorso 24 Giugno, Telegram ha rilasciato la nuova piattaforma dedicata ai bot (<https://telegram.org/blog/bot-revolution>). E' possibile ora migliorare l'esperienza utente tramite bot già disponibili oppure crearne uno personalizzato. La nuova piattaforma dispone di un accesso API tramite chiamate HTTP rendendo facile ed intuitivo l'interazione con i server di Telegram.

CodingJam usa i cookie per migliorare la tua esperienza di navigazione. Se continui a navigare su questo sito acconsenti all'uso dei cookie.

In questo articolo vedremo solo alcuni meccanismi base senza scendere nei dettagli per quanto riguarda il codice, per fornirti una idea su come strutturare il programma per un potenziale bot.

BotFather

Il BotFather, come il nome suggerisce, è il bot che vi aiuterà a generare nuovi bot. Attraverso un client Telegram si chatta con esso per la creazione di un bot e la generazione di un API token. Riporto una conversazione di esempio fatta con @BotFather (<https://telegram.me/botfather>).

Tu: /newbot

BotFather: Alright, a new bot. How are we going to call it? Please choose a name for your bot.

Tu: IlTuoBelBot

BotFather: Good. Now let's choose a username for your bot. It must end in `bot`. Like this, for example: TetrisBot or tetris_bot.

Tu: NomeBot

BotFather:

Done! Congratulations on your new bot. You will find it at telegram.me/NomeBot. You can now add a description, about section and profile picture for your bot, see /help for a list of commands.

Use this token to access the HTTP API:
125760859:AAE8Y15d20_Augr8WbFcs3_VT4R48u515lQ

For a description of the Bot API, see this page: <https://core.telegram.org/bots/api>

Bisogna prestare attenzione che l'username per il bot deve necessariamente terminare con la parola bot.

Interazione con il bot

Per interagire con il bot in teoria basterebbe utilizzare direttamente il proprio browser. Tuttavia per chiarezza utilizzeremo l'applicazione Postman (<https://www.getpostman.com/>). Ogni richiesta che andremo a fare sarà formattato secondo questo schema:

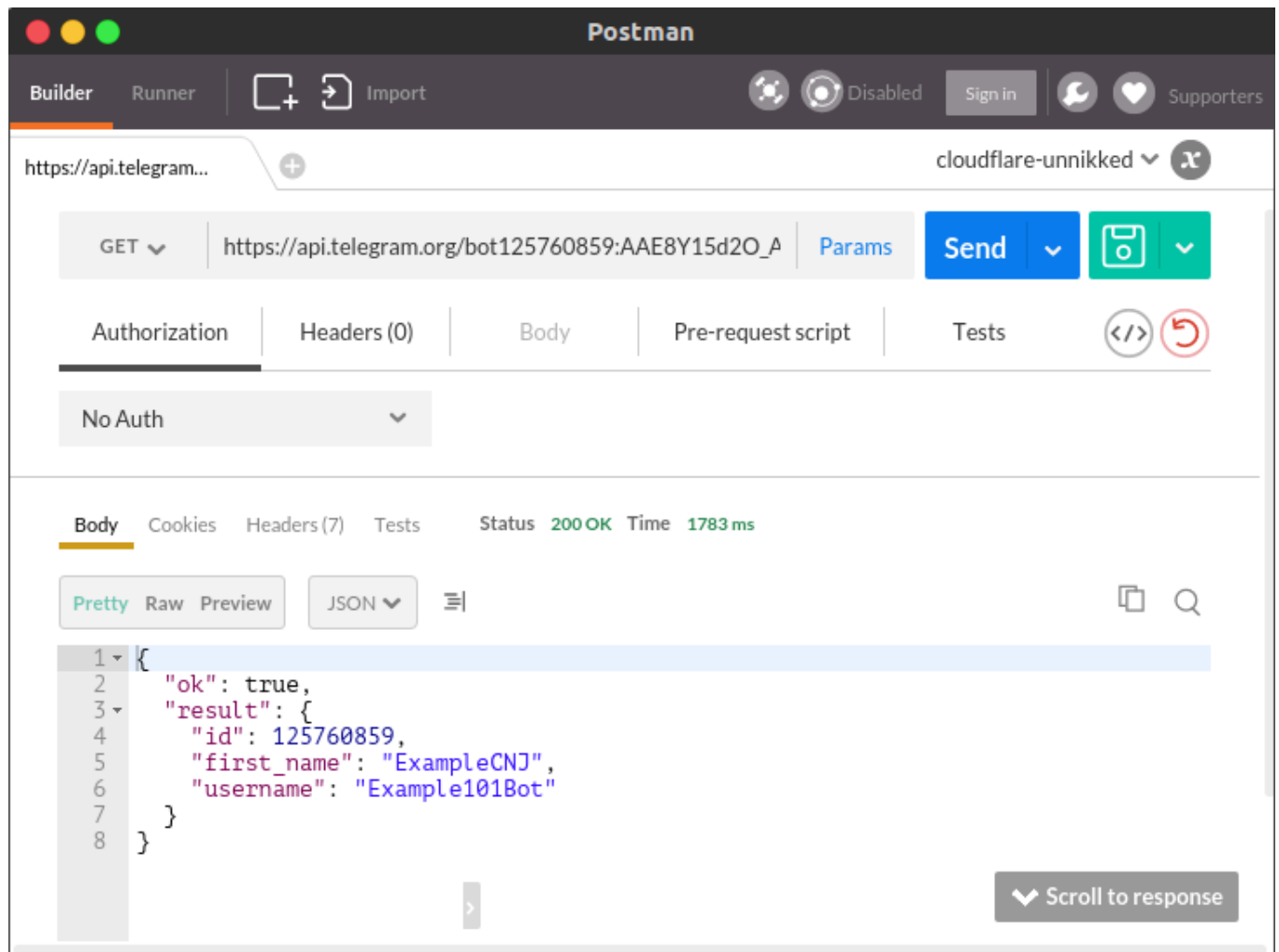
```
https://api.telegram.org/bot/METHOD_NAME
```

CodingJam usa i cookie per migliorare la tua esperienza di navigazione. Se continui a navigare su questo sito accetti l'uso dei cookie.

Dove `token` è il token che abbiamo generato chattando con @BotFather e `METHOD_NAME` è il metodo da far eseguire al bot. Ogni risposta da parte dei bot sarà formattato in JSON. Per una lista dei comandi supportati è consigliabile consultare la documentazione ufficiale (https://api.telegram.org/bot<token>/METHOD_NAME).

getMe

E' il comando che ci farà ottenere le informazioni base del proprio bot.



Ottenere l'input utente

Ci sono due modi per ottenere l'input da un bot cioè i messaggi che gli utenti inviano ad esso per l'interazione:

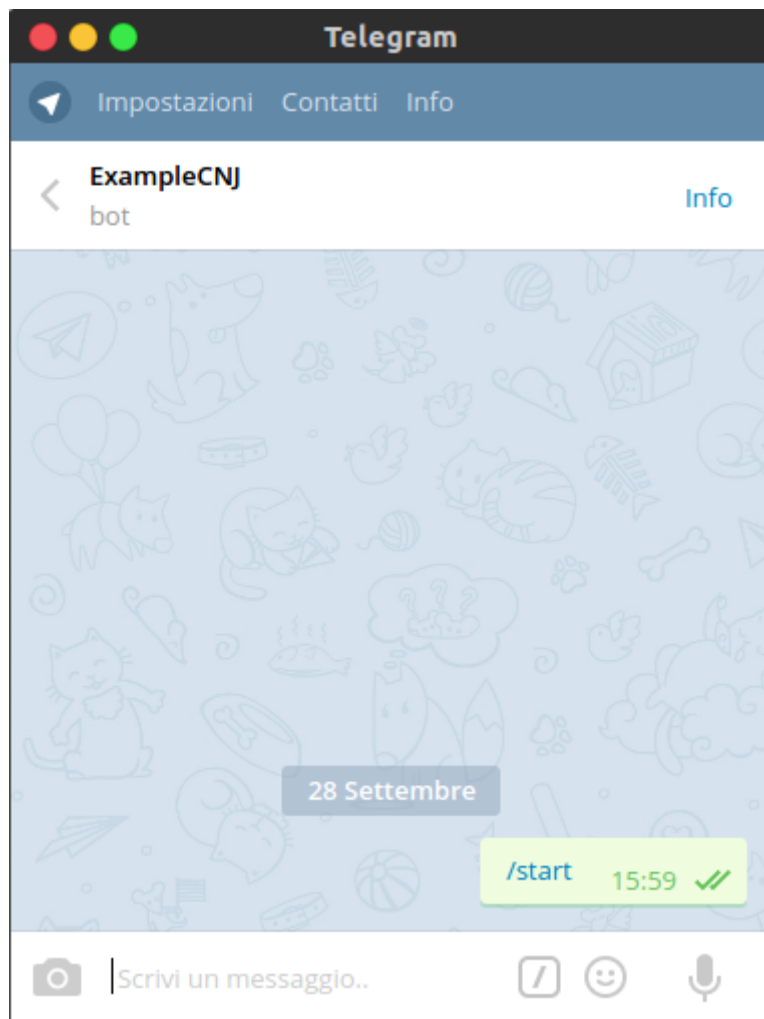
- Tramite la tecnica di long polling (https://en.wikipedia.org/wiki/Push_technology#Long_polling)
- Tramite la tecnica webhook (<https://en.wikipedia.org/wiki/Webhook>)

La tecnica long polling (metodo `getUpdates`) consiste nell'interrogare periodicamente i server di Telegram per determinare se ci sono messaggi da processare, useremo questo metodo (manualmente) per ottenere il campo `chat_id` necessario per l'interazione con uno specifico utente.

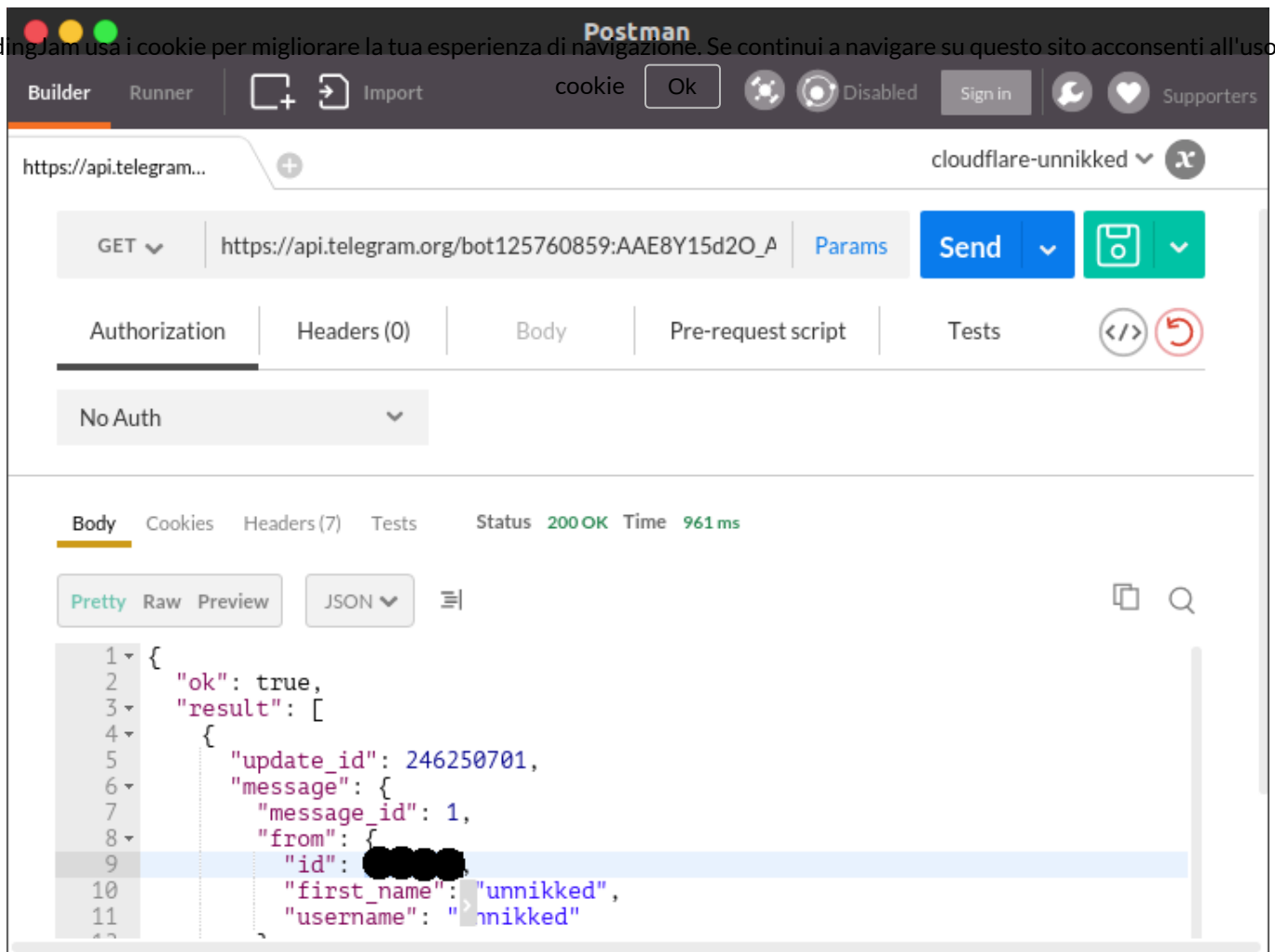
Tramite la tecnica **webhook** invece riceveremo automaticamente su una “callback” i nuovi messaggi da processare direttamente dai server di Telegram. Attraverso il **long polling** è il nostro codice che periodicamente deve interrogare i server di Telegram per ottenere nuove informazioni, avviene invece il contrario tramite l’uso di un **webhook**. Bisogna notare che i due metodi sono mutualmente esclusivi, una volta impostato una **callback** (tramite il metodo `setWebhook`) non è più possibile utilizzare il metodo `getUpdates`.

sendMessage

Per effettuare un test spediamoci da solo un messaggio. Per prima cosa dobbiamo utilizzare il nostro client per inviare un messaggio `/start` (è una convenzione) che servirà a instaurare un “handshake” tra un utente Telegram ed il bot. Un bot non può mai avviare una conversazione con un utente arbitrariamente.



Se lanciamo ora il metodo `getUpdates` tramite Postman otterremo qualcosa del genere.



Per chiarezza riporto la risposta in JSON

```
{
  "ok": true,
  "result": [
    {
      "update_id": 246250701,
      "message": {
        "message_id": 1,
        "from": {
          "id": 12345,
          "first_name": "unnikked",
          "username": "unnikked"
        }
      },
      "chat": {
        "id": 12345,
        "first_name": "unnikked",
        "username": "unnikked"
      },
      "date": 1443448777,
      "text": "/start"
    }
  ]
}
```

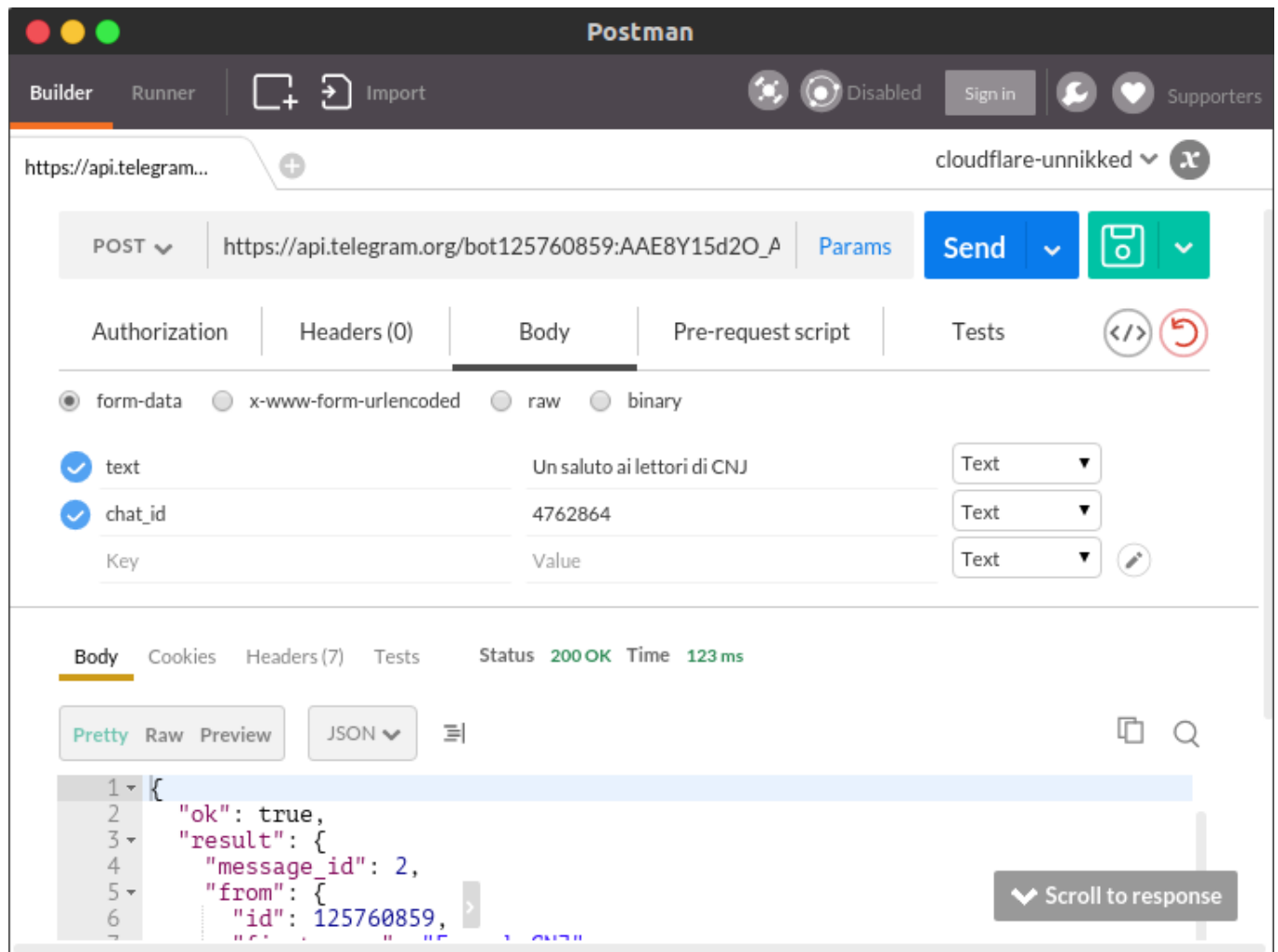
CodingJam usa i cookie per migliorare la tua esperienza di navigazione. Se continui a navigare su questo sito acconsenti all'uso dei cookie.

La proprietà `id` dell'oggetto `chat` è proprio l'id cliente da utilizzare nelle prossime chiamate HTTP per spedire messaggi o file multimediali.

Nel nostro caso di esempio è 12345. Da notare come nel campo text sia presente il messaggio `"/start"`.

sendMessage

Per spedire un messaggio è necessario utilizzare il metodo `sendMessage` che accetta come parametri il campo `text` e il campo `chat_id`. La richiesta può essere spedita sia utilizzando il metodo `POST` o `GET` HTTP.

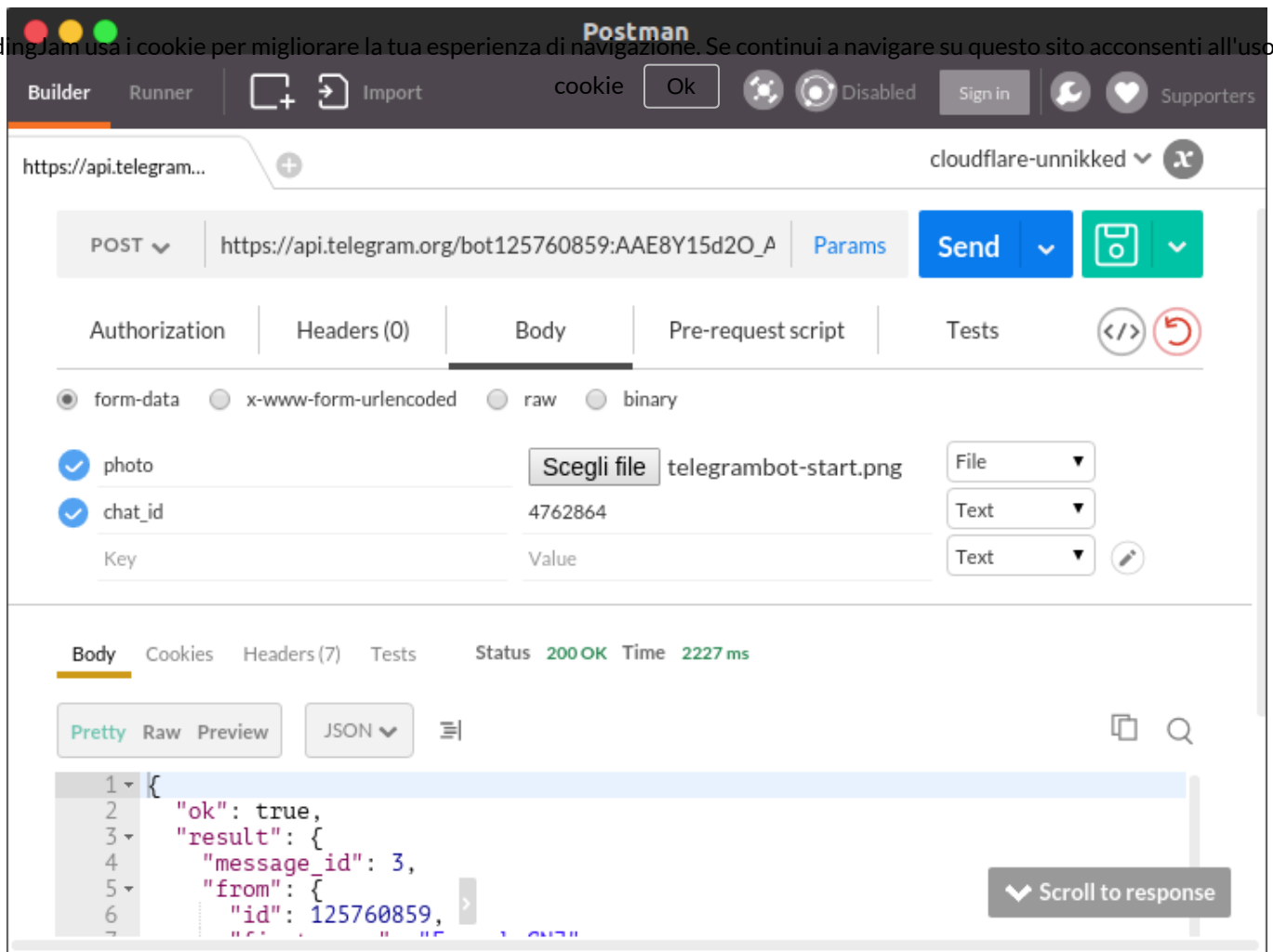


Che produrrà l'effetto di inviare un messaggio all'utente

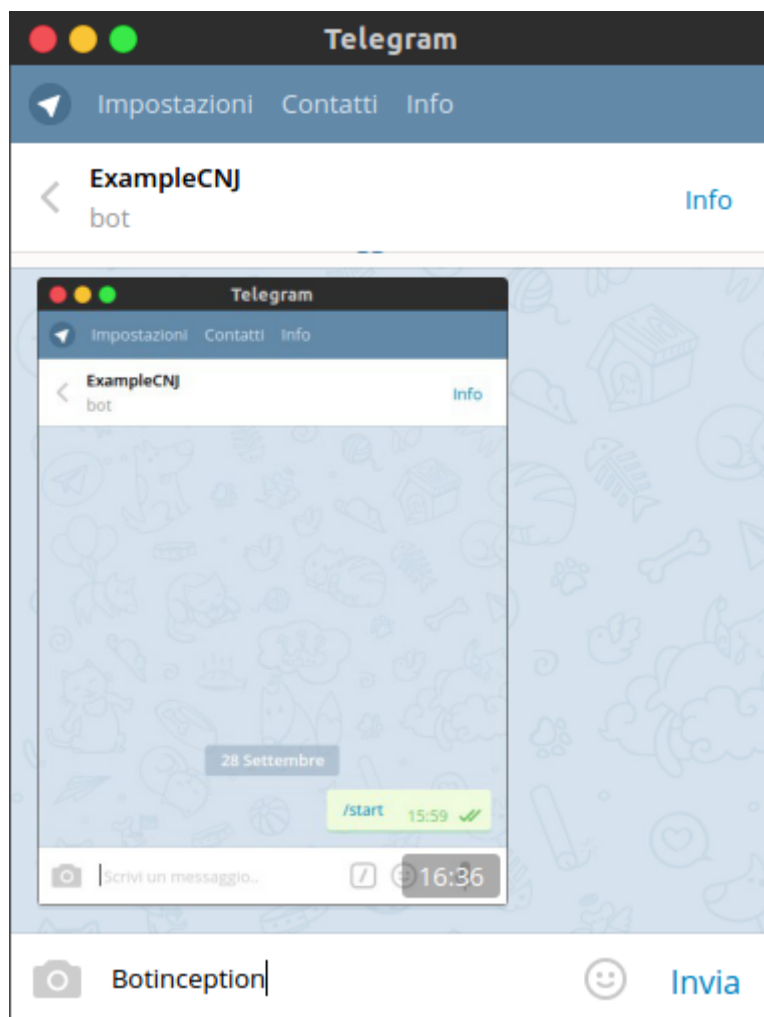


sendPhoto

Come ulteriore esempio vediamo come inviare una foto ad un utente tramite le API, bisogna utilizzare il metodo `sendPhoto`, utilizzando i campi `photo` e `chat_id`, ed inviare il form come `multipart/form-data` (Postman gestisce questo aspetto per voi).



Il risultato sarà



Conclusioni

La piattaforma bot, secondo il mio parere, è una buona idea poiché va a colmare alcune mancanze che le altre piattaforme non offrono.

I bot potrebbero essere usati nelle chat di gruppo sia per divertimento:

- @ImageBot (<https://telegram.me/imagebot>) – permette di cercare immagini attraverso keyword.
- @TriviaBot (<https://telegram.me/triviabot>) – permette di giocare a quiz trivia.
- @PoolBot (<https://telegram.me/pollbot>) – permette di creare sondaggi da condividere in altre chat.

O per la collaborazione in gruppi chat:

- @GithubBot (<https://telegram.me/githubbot>) – permette di ricevere notifiche da repository GitHub.

Anche per un uso personale sono indicati quei bot che aggiungono funzionalità alla piattaforma senza necessariamente essere supportate nativamente

- @AlertBot (<https://telegram.me/alertbot>) – permette di impostare promemoria direttamente nella chat.
- @HotOrBot (<https://telegram.me/hotorbot>) – permette di incontrare persone a caso.

In un prossimo articolo vedremo qualche esempio su come strutturare il codice di un bot.

Link utili

Qui di seguito vi propongo una lista di link utili tratti dal sito ufficiale di Telegram:

- Annuncio ufficiale del lancio della piattaforma Bot (<https://telegram.org/blog/bot-revolution>) sul blog di Telegram.
- Introduzione ai Bot (<https://core.telegram.org/bots>) fatta dal team di Telegram.
- Documentazione e novità (<https://core.telegram.org/bots/api>) delle API.
- FAQ (<https://core.telegram.org/bots/faq>) con codici di esempio.

Share story





Nicola Malizia

(http://codingjam.it/author/unnikked/)

Studente di Ingegneria informatica presso l'Università della Calabria.

Appassionato del mondo informatico fin da bambino, trascorre parte del suo tempo libero a sperimentare nuove tecnologie ed ad esplorare concetti della Computer Science. La condivisione è il suo obiettivo principale, mantiene due blog: unnikked.tk rivolto ad un pubblico esclusivamente italiano dove tratta tutorial pratici e semplici da seguire, con una impostazione da sistemista, unnikked.ga è rivolto invece ad un pubblico più ampio e tratta argomenti più complessi propri del suo campo di studi.

3 Commenti

Cose Non Javiste

cookie

 1 Accedi ▾

 Consiglia 3


 Condividi

Ordina dal migliore ▾



Partecipa alla discussione...

ENTRA CON

O REGISTRATI SU DISQUS 

Nome



Silvano Norberti • 2 mesi fa

Com'è possibile che Postman ti abbia dato una risposta quando in un punto l'url è <https://api.telegram.org/bo...> senza slash dopo "bot" e invece prima scrivi <https://api.telegram.org/bo...>

Poi il metodo getUpdates dove dovrebbe trovarsi su Postman? Dovresti essere più chiaro :/

^ | ▾ • Rispondi • Condividi >



Nicola Malizia ➔ Silvano Norberti • 15 giorni fa

Gli screenshot non mostrano l'url completo :) Il metodo getUpdates va utilizzato nel caso tu voglia implementare il long pooling. Postman è usato per mostrare alcuni esempi.

Se vuoi approfondire l'argomento può vedere altri articoli sul mio blog personale <https://unnikked.ga/search?...>

3 ^ | ▾ • Rispondi • Condividi >



Pietro • 2 anni fa

Come faccia a creare un "comando" ma senza mettere lo / all'inizio (es. lo~ciao bot~ciao, come va?)

In modo che se io scriva una parola anche senza lo / lui mi possa rispondere?

^ | ▾ • Rispondi • Condividi >

SEMPRE SU COSE NON JAVISTE

Ionic 2: prime impressioni

3 commenti • un anno fa

Alberto — Grazie, do un'occhiata all'articolo!

Screencast su Ionic: disponibile la registrazione

2 commenti • 2 anni fa

Francesco Strazzullo — Ciao Alessandro, sicuramente già adesso con cellulari Android di ultima generazione hai performance ...

Webpack ECMAScript 6 – Parte II

3 commenti • 8 mesi fa

Claudio Ferrari — Ciao, ottimo tutorial, complimenti. Vorrei sapere come aggiornare la parte start in package.json. Grazie

Introduzione ad Angular 2 – Parte 2

1 commento • un anno fa

DeepCode — Sinceramente devo fare i complimenti per l'articolo, ottima padronanza dell'argomento seppur il angular 2 sia

Scrivere su CodingJam



Vorresti scrivere un post su CodingJam?

SCOPRI COME

I tweet di CNJ

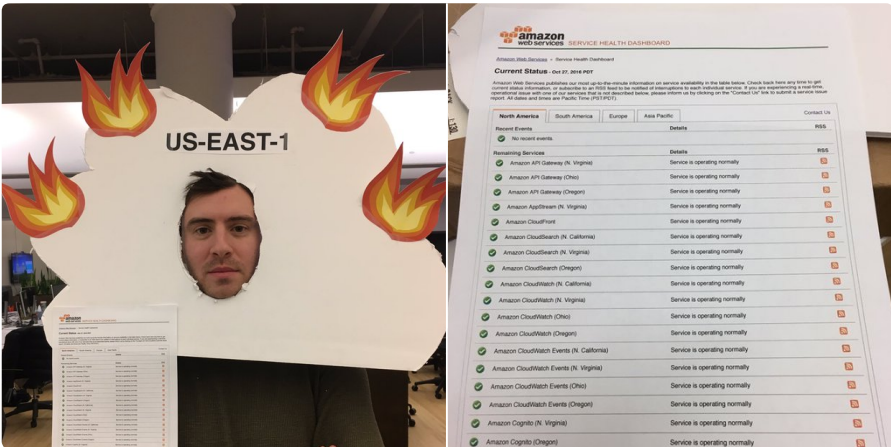
Tweets by @codingjam

CodingJam Retweeted



Justin Lintz @lintzston

My spooky Halloween costume **#AWS** **#cloud**



[Embed](#)

[View on Twitter](#)