



EBook Gratuito

APPENDIMENTO

Twilio

Free unaffiliated eBook created from
Stack Overflow contributors.

#twilio

Sommario

Di.....	1
Capitolo 1: Iniziare con Twilio.....	2
Osservazioni.....	2
Versioni.....	2
Examples.....	2
Invio di un messaggio SMS da un numero statunitense.....	2
Invio di un messaggio SMS tramite Promises.....	3
Capitolo 2: Come inviare un SMS con Ruby.....	5
Examples.....	5
Utilizzando la gemma Twilio.....	5
Capitolo 3: Come ricevere messaggi SMS con Ruby on Rails.....	6
Examples.....	6
Utilizzando la gemma Twilio.....	6
Capitolo 4: Invia messaggi in Python.....	8
Sintassi.....	8
Parametri.....	8
Osservazioni.....	8
Examples.....	8
Invio di MMS.....	8
Invio di SMS.....	9
Invio di messaggi e MMS tramite Twilio.....	9
Capitolo 5: Invia un messaggio SMS usando Bash e cURL.....	11
Sintassi.....	11
Parametri.....	11
Examples.....	11
Invia un singolo messaggio utilizzando un ID mittente alfanumerico.....	11
Capitolo 6: Invio di messaggi in Node.js.....	13
introduzione.....	13
Parametri.....	13
Osservazioni.....	13

Deprecazione dell'SDK 2.0	13
Riferimento ai parametri	13
Examples.....	14
Invio del tuo primo messaggio.....	14
Installazione del modulo del nodo.....	14
Invio del tuo messaggio.....	14
Titoli di coda	15

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [twilio](#)

It is an unofficial and free Twilio ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official Twilio.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capitolo 1: Iniziare con Twilio

Osservazioni

[Twilio](#) fornisce una piattaforma per la comunicazione tramite messaggistica IP (chat), video, voce, SMS e MMS. Questo semplice esempio dimostra come inviare un messaggio di testo SMS utilizzando il [modulo Twilio per Node.js](#).

Versioni

Versione	Data di rilascio
1.0.0	2016/04/12

Examples

Invio di un messaggio SMS da un numero statunitense

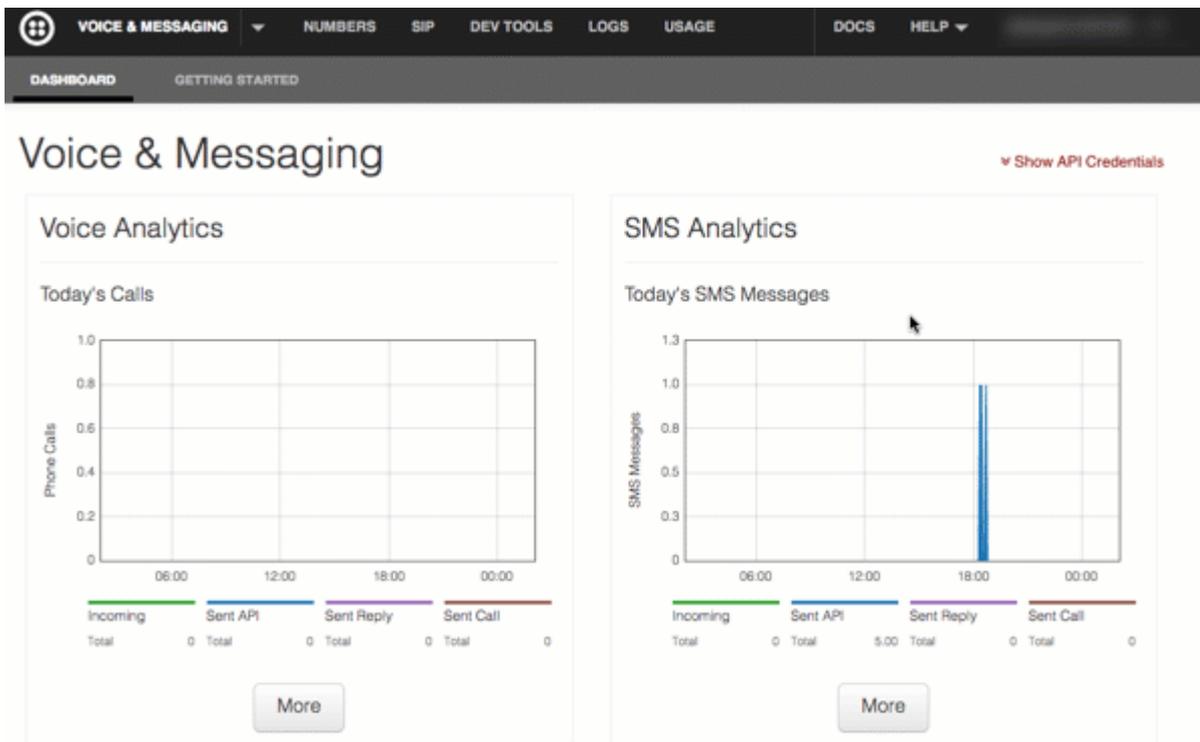
Ecco come inviare un messaggio di testo SMS da un numero statunitense usando l'SDK Node.js di Twilio.

Per prima cosa è necessario installare il client Node.js utilizzando:

```
npm install twilio
```

Quindi, devi [creare un account sul loro sito web](#) .

Una volta creato un account, è necessario il SID dell'account e il token di autenticazione che è possibile trovare nella dashboard online.



Nell'esempio di codice riportato di seguito, sostituisci `[Account SID]` e `[Auth Token]` Con quelli del tuo account.

```
// Twilio Credentials
var accountSid = '[Account SID]';
var authToken = '[Auth Token]';

//require the Twilio module and create a REST client
var client = require('twilio')(accountSid, authToken);

client.messages.create({
  to: "+16518675309", // Any number Twilio can deliver to
  from: "+14158141829", // A number you bought from Twilio and can use for outbound
  communication
  body: "Hey Jenny, thanks for the pull request, will merge it right away."
}, function(err, message) {
  console.log(message.sid);
});
```

Invio di un messaggio SMS tramite Promises

L'API Node.JS di Twilio supporta in modo nativo le promesse, consentendo di utilizzare le promesse per l'invio di messaggi SMS (questo esempio è stato preso e adattato direttamente dai [documenti API di Twilio](#)).

```
// Create an authenticated Twilio REST API client
var twilio = require('twilio');
var client = new twilio.RestClient('ACCOUNT_SID', 'AUTH_TOKEN');

// A simple example of sending an sms message using promises
var promise = client.makeCall({
  to: '+16515556667777', // a number to call
  from: '+16518889999', // a Twilio number you own
```

```
    body: 'Hello, world.' // A URL containing TwiML instructions for the call
  });

// You can assign functions to be called, at any time, after the request to
// Twilio has been completed. The first function is called when the request
// succeeds, the second if there was an error.
promise
  .then(function(sms) {
    console.log('Call success! SMS SID: ' + sms.sid);
  }, function(error) {
    console.error('Call failed! Reason: ' + error.message);
  });
```

Leggi Iniziare con Twilio online: <https://riptutorial.com/it/twilio/topic/443/iniziare-con-twilio>

Capitolo 2: Come inviare un SMS con Ruby

Examples

Utilizzando la gemma Twilio

Questo presuppone che tu abbia un account twilio e hai acquistato / prenotato un numero di telefono ...

Se stai usando bundler aggiungi la seguente riga al tuo Gemfile per includere `twilio-ruby` nel tuo progetto:

```
gem 'twilio-ruby'
```

altrimenti inserisci `gem install twilio-ruby` nella riga di comando. Potresti aver bisogno di `sudo` se stai usando il sistema Ruby e non `ruby-env` ecc.

e il codice

```
# Instantiate a Twilio REST client with your account SID and auth token,  
# which can be found on your dashboard.  
  
client = Twilio::REST::Client.new("your_account_sid", "your_auth_token")  
  
client.messages.create(from: "+15556667777", to: "+15558883333", body: "Hello, world!")
```

Leggi Come inviare un SMS con Ruby online: <https://riptutorial.com/it/twilio/topic/1760/come-inviare-un-sms-con-ruby>

Capitolo 3: Come ricevere messaggi SMS con Ruby on Rails

Examples

Utilizzando la gemma Twilio

Supponendo di avere un account Twilio e [credenziali API](#) , aggiungi quanto segue al tuo Gemfile:

```
gem 'twilio-ruby'
```

In alternativa puoi `gem install twilio-ruby` .

Per fare in modo che Twilio invii un SMS in arrivo a un determinato percorso nell'applicazione, è necessario configurare l'URL di messaggistica per il proprio [numero di telefono](#) . Una volta fatto, è necessario impostare una rotta in `config/routes.rb` :

```
TwilioSample::Application.routes.draw do
  resources :messages
end
```

Questo crea un set RESTful routes (`GET /messages` per elencare messaggi, `POST /messages` per creare un messaggio, ecc.) Che invierà richieste al `MessagesController` . Definiti questi percorsi, si imposta l'URL di messaggistica nella bacheca di Twilio su `https://your.site/messages` come WebHook HTTP POST.

Il tuo controller gestisce il messaggio:

```
class MessagesController < ApplicationController

  def create
    # First, check to see if this message really came from Twilio
    uri, signature = [request.protocol, request.host_with_port, request.path].join,
    request.env['HTTP_X_TWILIO_SIGNATURE']
    validator = Twilio::Util::RequestValidator.new(ENV['TWILIO_AUTH_TOKEN'])

    if validator.validate(uri, request.request_parameters, signature)
      # The params hash has the data from the Twilio payload
      Rails.logger.info "I just got this message from #{params[:From]}: #{params[:Body]}"

      # Send back a response that Twilio can handle
      render xml: { Sms: "Thanks, I got your message" }.to_xml(root: 'Response')

    else
      # This request didn't come from Twilio, so return a "Forbidden" response
      head(403)
    end
  end

end
```

end

Leggi Come ricevere messaggi SMS con Ruby on Rails online:

<https://riptutorial.com/it/twilio/topic/3673/come-ricevere-messaggi-sms-con-ruby-on-rails>

Capitolo 4: Invia messaggi in Python

Sintassi

- classe `twilio.rest.resources.Messages (* args, ** kwargs)`

Parametri

Parametro	Dettagli
<code>a (str)</code>	Il numero di telefono di destinazione.
<code>da_ (str)</code>	Il numero di telefono che invia questo messaggio (deve essere un numero Twilio verificato)
<code>corpo (str)</code>	Il messaggio che si desidera inviare, limitato a 160 caratteri.
<code>status_callback</code>	Un URL su cui Twilio eseguirà il POST quando il tuo messaggio verrà elaborato.
<code>application_sid (str)</code>	Il carattere a 34 caratteri dell'applicazione Twilio deve essere utilizzato per gestire questa chiamata telefonica.

Osservazioni

Twilio invierà messaggi SMS per codici lunghi (negli Stati Uniti, un numero di dieci cifre normale) alla velocità di un messaggio al secondo. Per un ulteriore throughput, è possibile utilizzare pool di numeri con [Copilot](#) o richiedere uno [shortcode](#) , che può inviare 30 messaggi o più al secondo.

Examples

Invio di MMS

Puoi aggiungere un'immagine al tuo messaggio usando il parametro `media_url` .

```
# Download the twilio-python library from http://twilio.com/docs/libraries
from twilio.rest import TwilioRestClient

# Find these values at https://twilio.com/user/account
account_sid = "ACXXXXXXXXXXXXXXXXXXXX"
auth_token = "YYYYYYYYYYYYYYYYYYYY"
client = TwilioRestClient(account_sid, auth_token)
message = client.messages.create(
    to="+12316851234",
    from_="TWILIO_NUMBER",
    body="Hello there, StackOverflow!",
```

```
media_url=[
    'https://demo.twilio.com/owl.png',
    'https://demo.twilio.com/logo.png'])
```

Invio di SMS

Per inviare il tuo primo SMS con Twilio e Python avrai solo bisogno della [libreria di helper Twilio-Python](#) per iniziare.

```
# Download the twilio-python library from http://twilio.com/docs/libraries
from twilio.rest import TwilioRestClient

# Find these values at https://twilio.com/user/account
account_sid = "ACXXXXXXXXXXXXXXXXXXXX"
auth_token = "YYYYYYYYYYYYYYYYYYYY"
client = TwilioRestClient(account_sid, auth_token)

message = client.messages.create(to="+12316851234", from_="TWILIO_NUMBER",
                                body="Hello there, StackOverflow!")
```

Invio di messaggi e MMS tramite Twilio

Twilio aiuta a creare app che comunichino con chiunque nel mondo. API voce e video, messaggistica e autenticazione per ogni applicazione.

È possibile ottenere una chiave API gratuitamente.

Per inviare un messaggio tramite Twilio, la tua applicazione deve fare una richiesta POST HTTP a Twilio con le seguenti cose:

1. Il numero di telefono a cui si desidera inviare il messaggio.
2. Numero Twilio da cui stai tentando di inviare il messaggio. (Funzionano solo i numeri di telefono abilitati per i messaggi di Twilio)
3. Il corpo del messaggio.

Per inviare un SMS, effettuare una richiesta POST HTTP alla risorsa Messaggi.

```
POST https://api.twilio.com/20xx-xx-xx/Accounts/xxxxxxx/Messages
```

Di seguito è riportato un codice di esempio per mostrare come funzionerà l'invio di messaggi con l'API Twilio.

```
# Download the twilio-python library from http://twilio.com/docs/libraries
from twilio.rest import TwilioRestClient

# Find these values at https://twilio.com/user/account
account_sid = "ACXXXXXXXXXXXXXXXXXXXX"
auth_token = "YYYYYYYYYYYYYYYYYYYY"
client = TwilioRestClient(account_sid, auth_token)

message = client.messages.create(
```

```
to      = "+12316851234",
from_   = "+15555555555",
body    = "Hello there!"
)
```

Se il numero Twilio supporta MMS, è possibile anche inviare e ricevere MMS.

Di seguito è riportato il codice per inviare MMS tramite Twilio API.

```
message = client.messages.create(
    to      = "+12316851234",
    from_   = "+15555555555",
    body    = "Hello there!",
    media_url=[
        'https://demo.twilio.com/owl.png',
        'https://demo.twilio.com/logo.png'
    ])
```

Leggi [Invia messaggi in Python online](https://riptutorial.com/it/twilio/topic/579/invia-messaggi-in-python): <https://riptutorial.com/it/twilio/topic/579/invia-messaggi-in-python>

Capitolo 5: Invia un messaggio SMS usando Bash e cURL

Sintassi

- [https://api.twilio.com/ {Versione Api} / Account / {AccountSid} /Messages.json](https://api.twilio.com/{Versione Api} / Account / {AccountSid} /Messages.json) {corpo}

Parametri

Parametro	Dettagli
{Versione Api}	Data di versione dell'API. ad esempio 2010-04-01
{Account Sid}	L'account Sid. Inizia AC
{corpo}	Corpo con codifica URL compreso Da, A, Corpo e MessagingServiceSid

Examples

Invia un singolo messaggio utilizzando un ID mittente alfanumerico

Una richiesta POST HTTP viene inviata a un URL del formato: " [https://api.twilio.com/2xxx-xx-xx/Accounts/\[AccountSid\]/Messages.json](https://api.twilio.com/2xxx-xx-xx/Accounts/[AccountSid]/Messages.json)

L'esempio seguente utilizza una stringa alfanumerica come mittente. Al momento della stesura di un ID mittente può essere aggiunto solo tramite una richiesta di servizio Twilio.

Richiesta di esempio:

```
To="+447111111111" ;
From="Wxxxxxxxx" ;
MessagingServiceSid="MGxxxxxxxxxxxxxxxxxxxxxxxx" ;
Body="Test Message" ;
AccountSid="ACxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx" ;
AuthToken="[Auth Token]" ;
CMD="curl --request POST \"https://api.twilio.com/2010-04-01/Accounts/${AccountSid?}/Messages.json\" \" \" ;
CMD=\"${CMD?}\"--data-urlencode \"From=${From?}\" \" \" ;
CMD=\"${CMD?}\"--data-urlencode \"To=${To?}\" \" \" ;
CMD=\"${CMD?}\"--data-urlencode \"MessagingServiceSid=${MessagingServiceSid?}\" \" \" ;
CMD=\"${CMD?}\"--data-urlencode \"Body=${Body?}\" \" \" ;
CMD=\"${CMD?}\"-u \"${AccountSid?}:${AuthToken?}\" \" \" ;
echo \"${CMD?}\" ;
eval \"${CMD?}\" ;
```

Risposta di esempio (JSON, formattata):

```
{
  "sid": "SMxxxxxxxxxxxx",
  "date_created": "Tue, 26 Jul 2016 12:50:07 +0000",
  "date_updated": "Tue, 26 Jul 2016 12:50:07 +0000",
  "date_sent": null,
  "account_sid": "ACxxxxxxxx",
  "to": "+44711111111",
  "from": "Wxxxxxxxx",
  "messaging_service_sid": "MGxxxxxxxx",
  "body": "Test Message",
  "status": "accepted",
  "num_segments": "0",
  "num_media": "0",
  "direction": "outbound-api",
  "api_version": "2010-04-01",
  "price": null,
  "price_unit": null,
  "error_code": null,
  "error_message": null,
  "uri": "/2010-04-01/Accounts/ACxxxxxxxx/Messages/SMxxxxxxxx.json",
  "subresource_uris": {
    "media": "/2010-04-01/Accounts/ACxxxxxxxx/Messages/SMxxxxxxxx/Media.json"
  }
}
```

[Leggi Invia un messaggio SMS usando Bash e cURL online:](https://riptutorial.com/it/twilio/topic/4237/invia-un-messaggio-sms-usando-bash-e-curl)

<https://riptutorial.com/it/twilio/topic/4237/invia-un-messaggio-sms-usando-bash-e-curl>

Capitolo 6: Invio di messaggi in Node.js

introduzione

Twilio consente di inviare e ricevere messaggi di testo con Node.js utilizzando la libreria del nodo twilio per effettuare chiamate. Questa guida presuppone che tu abbia già configurato un account con Twilio e che abbia il tuo SID dell'account e il token di autenticazione dalla Console Twilio.

Parametri

Parametro	Dettagli
a	Un numero di telefono valido per ricevere il messaggio
a partire dal	Un numero Twilio che ti è stato assegnato
corpo	Il corpo del messaggio di testo è limitato a 1600 caratteri
StatusCallback	Un URL che Twilio invia quando cambia lo stato di un messaggio
MAXPRICE	Imposta il prezzo massimo di un messaggio
periodo di validità	Il numero di secondi in cui il messaggio rimane nella coda Twilio
fornire un feedback	Valore booleano, se impostato su true
mediaUrl	Un URL contenente un contenuto gif, png o jpeg che verrà inviato con il messaggio

Osservazioni

Deprecazione dell'SDK 2.0

Twilio ha due versioni del modulo twilio-node, un 2.0 SDK e un 3.0 SDK. Si consiglia di utilizzare l'SDK 3.0 in quanto l'SDK 2.0 verrà ritirato l'8/31/2017.

Avviso di cancellazione: la nuova funzionalità verrà aggiunta alla nuova libreria (Node Helper Library 3.x). La vecchia biblioteca (2.x) sarà ufficialmente supportata fino al 31/08/2017. Dopo quel giorno, Twilio smetterà di fornire correzioni di bug e il Supporto potrebbe chiederti di aggiornare prima di risolvere i problemi.

<https://www.twilio.com/docs/libraries/node>

Riferimento ai parametri

È possibile fare riferimento ai [documenti REST di Twilio](#) per una descrizione più dettagliata. Twilio ha anche un [JSdocs](#) che può essere usato come riferimento.

Examples

Invio del tuo primo messaggio

Installazione del modulo del nodo

È possibile installare questo modulo eseguendo il comando seguente nella directory del progetto:

```
npm install twilio
```

Invio del tuo messaggio

```
const accountSID = ''; // Obtained from the Twilio Console
const authToken = ''; // Obtained from the Twilio Console

const client = require('twilio')(accountSID, authToken);

client.messages.create({
  to: '', // Number you want to send to
  from: '', // From a valid Twilio number
  body: 'Hello, from Stack Overflow', // What appears in the text message
})
.then((message) => console.log(message.sid));
```

Leggi [Invio di messaggi in Node.js online](https://riptutorial.com/it/twilio/topic/9955/invio-di-messaggi-in-node-js): <https://riptutorial.com/it/twilio/topic/9955/invio-di-messaggi-in-node-js>

Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con Twilio	Adam , cdock , Community , ddoria921 , Kevin Whinnery , msmanek , Sven
2	Come inviare un SMS con Ruby	derrek , Steve Graham
3	Come ricevere messaggi SMS con Ruby on Rails	Benjamin Curtis
4	Invia messaggi in Python	anthon , Henk Langeveld , Jacob Budin , Megan Speir , Shubham Gupta
5	Invia un messaggio SMS usando Bash e cURL	antonycc
6	Invio di messaggi in Node.js	Matt Shirley