



EBook Gratis

APRENDIZAJE

Twilio

Free unaffiliated eBook created from
Stack Overflow contributors.

#twilio

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con Twilio.....	2
Observaciones.....	2
Versiones.....	2
Examples.....	2
Enviar un mensaje SMS desde un número de Estados Unidos.....	2
Enviando un mensaje SMS usando promesas.....	3
Capítulo 2: Cómo enviar un SMS con Ruby.....	5
Examples.....	5
Usando la Gema Twilio.....	5
Capítulo 3: Cómo recibir mensajes SMS con Ruby on Rails.....	6
Examples.....	6
Usando la Gema Twilio.....	6
Capítulo 4: Enviando mensajes en Node.js.....	8
Introducción.....	8
Parámetros.....	8
Observaciones.....	8
Depredación 2.0 SDK.....	8
Referencia de parámetros.....	8
Examples.....	9
Enviando su primer mensaje.....	9
Instalación del módulo de nodo.....	9
Enviando tu mensaje.....	9
Capítulo 5: Enviar mensajes en Python.....	10
Sintaxis.....	10
Parámetros.....	10
Observaciones.....	10
Examples.....	10
Enviando MMS.....	10
Enviando SMS.....	11

Enviando mensajes y MMS a través de Twilio.....	11
Capítulo 6: Enviar un mensaje SMS usando Bash y cURL.....	13
Sintaxis.....	13
Parámetros.....	13
Examples.....	13
Enviar un solo mensaje utilizando un ID de remitente alfanumérico.....	13
Creditos	15

Acerca de

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

Capítulo 1: Empezando con Twilio

Observaciones

Twilio proporciona una plataforma para la comunicación a través de mensajería IP (chat), video, voz, SMS y MMS. Este sencillo ejemplo muestra cómo enviar un mensaje de texto SMS mediante el [módulo Twilio para Node.js](#).

Versiones

Versión	Fecha de lanzamiento
1.0.0	2016-04-12

Examples

Enviar un mensaje SMS desde un número de Estados Unidos

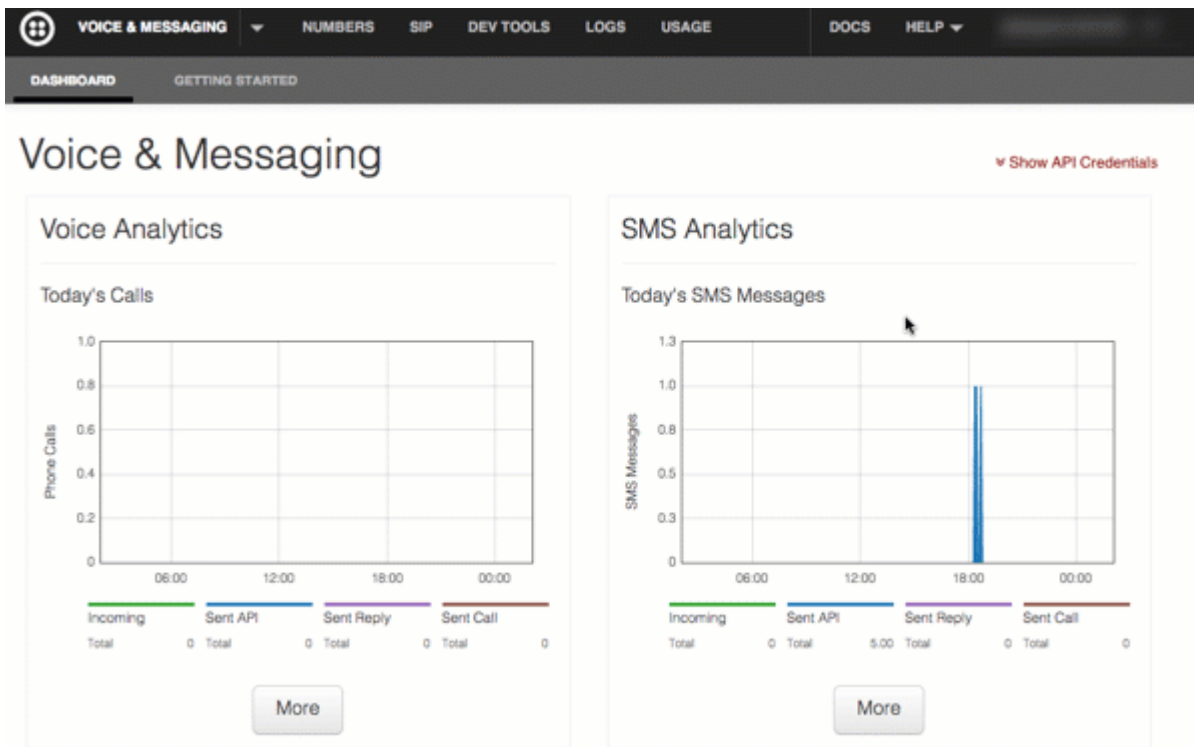
Esta es la forma de enviar un mensaje de texto SMS desde un número de EE. UU. Utilizando el SDK Node.js de Twilio.

Primero necesita instalar el cliente Node.js usando:

```
npm install twilio
```

Entonces, tienes que [crear una cuenta en su sitio web](#) .

Una vez que tenga una cuenta, necesitará el SID de la cuenta y el token de autenticación que puede encontrar en el panel de control en línea.



En el ejemplo de código a continuación, reemplace [Account SID] y [Auth Token] con los de su cuenta.

```
// 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);
});
```

Enviando un mensaje SMS usando promesas

La API Node.JS de Twilio admite promesas de forma nativa, lo que le permite usar promesas al enviar mensajes SMS (este ejemplo se tomó y adaptó directamente de [los documentos API de 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);
  });
```

Lea Empezando con Twilio en línea: <https://riptutorial.com/es/twilio/topic/443/empezando-con-twilio>

Capítulo 2: Cómo enviar un SMS con Ruby

Examples

Usando la Gema Twilio

Esto supone que tiene una cuenta de twilio y ha comprado / reservado un número de teléfono ...

Si está utilizando bundler, agregue la siguiente línea a su Gemfile para incluir `twilio-ruby` en su proyecto:

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

de lo contrario, ingrese `gem install twilio-ruby` en la línea de comando. Es posible que necesite `sudo` si está utilizando el sistema Ruby y no Ruby-env, etc.

y el código

```
# 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!")
```

Lea Cómo enviar un SMS con Ruby en línea: <https://riptutorial.com/es/twilio/topic/1760/como-enviar-un-sms-con-ruby>

Capítulo 3: Cómo recibir mensajes SMS con Ruby on Rails

Examples

Usando la Gema Twilio

Suponiendo que tiene una cuenta Twilio y [credenciales API](#) , agregue lo siguiente a su Gemfile:

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

Alternativamente puedes `gem install twilio-ruby` .

Para que Twilio envíe un SMS entrante a una ruta particular en su aplicación, debe configurar la URL de mensajería para su [número de teléfono](#) . Una vez hecho esto, debe configurar una ruta en `config/routes.rb` :

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

Esto crea un conjunto de rutas REST (`GET /messages` para listar mensajes, `POST /messages` para crear un mensaje, etc.) que enviarán solicitudes al `MessagesController` . Con estas rutas definidas, establecería la URL de mensajería en el panel de control de Twilio en `https://your.site/messages` como un HTTP POST Webhook.

Su controlador maneja el mensaje:

```
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
```

Lea Cómo recibir mensajes SMS con Ruby on Rails en línea:

<https://riptutorial.com/es/twilio/topic/3673/como-recibir-mensajes-sms-con-ruby-on-rails>

Capítulo 4: Enviando mensajes en Node.js

Introducción

Twilio le permite enviar y recibir mensajes de texto con Node.js usando la biblioteca de twilio-node para hacer llamadas. Esta guía asume que ya ha configurado una cuenta con Twilio y tiene el SID de su cuenta y el token de autenticación de la Consola Twilio.

Parámetros

Parámetro	Detalles
a	Un número de teléfono válido para recibir el mensaje.
desde	Un número de Twilio que se te asigna.
cuerpo	El cuerpo del mensaje de texto limitado a 1600 caracteres
StatusCallback	Una URL que Twilio publica cuando cambia el estado de un mensaje.
precio máximo	Establecer el precio máximo de un mensaje.
Período de validez	La cantidad de segundos que el mensaje permanecerá en la cola de Twilio.
suministre realimentación	Valor booleano, cuando se establece en verdadero
mediaUrl	Una URL que contiene un contenido gif, png o jpeg que se enviará con el mensaje

Observaciones

Depredación 2.0 SDK

Twilio tiene dos versiones del módulo twilio-node, un SDK 2.0 y un SDK 3.0. Se recomienda utilizar el SDK 3.0, ya que el SDK 2.0 quedará en desuso el 8/31/2017.

Aviso de eliminación: la nueva funcionalidad solo se agregará a la nueva biblioteca (Node Helper Library 3.x). La biblioteca antigua (2.x) se admitirá oficialmente hasta el 8/31/2017. Después de ese día, Twilio dejará de proporcionar correcciones de errores y es posible que el Soporte le pida que actualice antes de solucionar problemas.

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

Referencia de parámetros

Puede [consultar los documentos REST de Twilio](#) para obtener una descripción más detallada. Twilio también tiene un [JSdocs](#) que puede usarse como referencia.

Examples

Enviando su primer mensaje

Instalación del módulo de nodo

Puede instalar este módulo ejecutando el siguiente comando en el directorio de su proyecto:

```
npm install twilio
```

Enviando tu mensaje

```
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));
```

Lea [Enviando mensajes en Node.js en línea: https://riptutorial.com/es/twilio/topic/9955/enviando-mensajes-en-node-js](https://riptutorial.com/es/twilio/topic/9955/enviando-mensajes-en-node-js)

Capítulo 5: Enviar mensajes en Python

Sintaxis

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

Parámetros

Parámetro	Detalles
<code>a (str)</code>	El número de teléfono de destino.
<code>desde_ (str)</code>	El número de teléfono que envía este mensaje (debe ser un número de Twilio verificado)
<code>cuerpo (str)</code>	El mensaje que desea enviar, limitado a 160 caracteres.
<code>status_callback</code>	Una URL que Twilio publicará cuando su mensaje sea procesado.
<code>application_sid (str)</code>	El sid de 34 caracteres de la aplicación que Twilio debe usar para manejar esta llamada telefónica.

Observaciones

Twilio enviará mensajes SMS para códigos largos (en los EE. UU., Un número normal de diez dígitos) a una velocidad de un mensaje por segundo. Para obtener un rendimiento adicional, puede usar grupos de números con [Copilot](#) o solicitar un [shortcode](#) , que puede enviar 30 mensajes o más por segundo.

Examples

Enviando MMS

Puede agregar una imagen a su mensaje usando el parámetro `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'])
```

Enviando SMS

Para enviar tus primeros SMS con Twilio y Python, solo necesitarás la biblioteca auxiliar [Twilio-Python](#) para comenzar.

```
# 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!")
```

Enviando mensajes y MMS a través de Twilio.

Twilio ayuda a construir aplicaciones que se comunican con todos en el mundo. API de voz y video, mensajería y autenticación para cada aplicación.

Puede obtener una clave API de forma gratuita.

Para enviar un mensaje a través de Twilio, su aplicación debe realizar una solicitud HTTP POST a Twilio con las siguientes cosas:

1. El número de teléfono al que desea enviar el mensaje.
2. Número de twilio desde el que está intentando enviar el mensaje. (Solo funcionarán los números de teléfono habilitados para mensajes de Twilio)
3. El cuerpo del mensaje.

Para enviar un SMS, realice una solicitud HTTP POST al recurso de Mensajes.

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

A continuación se muestra un código de ejemplo para mostrar cómo funcionará el envío de mensajes con Twilio API.

```
# 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!"
)
```

Si el número de Twilio admite MMS, también puede enviar y recibir MMS.

A continuación se muestra el código para enviar MMS a través de 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'
    ])
)
```

Lea [Enviar mensajes en Python en línea: https://riptutorial.com/es/twilio/topic/579/Enviar-mensajes-en-python](https://riptutorial.com/es/twilio/topic/579/Enviar-mensajes-en-python)

Capítulo 6: Enviar un mensaje SMS usando Bash y cURL

Sintaxis

- [https://api.twilio.com/ {versión de Api} / Accounts / {AccountSid} /Messages.json](https://api.twilio.com/{versión de Api} / Accounts / {AccountSid} /Messages.json) {body}

Parámetros

Parámetro	Detalles
{Versión api}	Fecha de versión de la API. por ejemplo, 2010-04-01
{Cuenta Sid}	La Cuenta Sid. Inicia AC
{cuerpo}	Cuerpo codificado de URL que incluye De, A, Cuerpo y Servicio de mensajería

Examples

Enviar un solo mensaje utilizando un ID de remitente alfanumérico

Se envía una solicitud HTTP POST a una URL con el formato: " <https://api.twilio.com/2xxx-xx-xx/Accounts/{AccountSid}/Messages.json>

El siguiente ejemplo utiliza una cadena alfanumérica como remitente. En el momento de escribir un ID de remitente solo se puede agregar a través de una solicitud de servicio Twilio.

Solicitud de ejemplo:

```
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?}\" ;
```


Ejemplo de respuesta (JSON, formateado):

```
{
  "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": "ACxxxxxxxxxxxx",
  "to": "+447111111111",
  "from": "Wxxxxxxxx",
  "messaging_service_sid": "MGxxxxxxxxxxxx",
  "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/ACxxxxxxxxxxxx/Messages/SMxxxxxxxxxxxx.json",
  "subresource_uris": {
    "media": "/2010-04-01/Accounts/ACxxxxxxxxxxxx/Messages/SMxxxxxxxxxxxx/Media.json"
  }
}
```

Lea [Enviar un mensaje SMS usando Bash y cURL en línea:](https://riptutorial.com/es/twilio/topic/4237/enviar-un-mensaje-sms-usando-bash-y-curl)

<https://riptutorial.com/es/twilio/topic/4237/enviar-un-mensaje-sms-usando-bash-y-curl>

Creditos

S. No	Capítulos	Contributors
1	Empezando con Twilio	Adam , cdock , Community , ddoria921 , Kevin Whinnery , msmanek , Sven
2	Cómo enviar un SMS con Ruby	derrek , Steve Graham
3	Cómo recibir mensajes SMS con Ruby on Rails	Benjamin Curtis
4	Enviando mensajes en Node.js	Matt Shirley
5	Enviar mensajes en Python	anthon , Henk Langeveld , Jacob Budin , Megan Speir , Shubham Gupta
6	Enviar un mensaje SMS usando Bash y cURL	antonycc