

 eBook Gratuit

# APPRENEZ

---

# Twilio

eBook gratuit non affilié créé à partir des  
**contributeurs de Stack Overflow.**

#twilio

# Table des matières

À propos.....	1
<b>Chapitre 1: Démarrer avec Twilio.....</b>	<b>2</b>
Remarques.....	2
Versions.....	2
Exemples.....	2
Envoi d'un message SMS à partir d'un numéro américain.....	2
Envoi d'un message SMS à l'aide de promesses.....	3
<b>Chapitre 2: Comment envoyer un SMS avec Ruby.....</b>	<b>5</b>
Exemples.....	5
Utilisation du Twilio Gem.....	5
<b>Chapitre 3: Comment faire pour recevoir des messages SMS avec Ruby on Rails.....</b>	<b>6</b>
Exemples.....	6
Utilisation du Twilio Gem.....	6
<b>Chapitre 4: Envoi de messages dans Node.js.....</b>	<b>8</b>
Introduction.....	8
Paramètres.....	8
Remarques.....	8
<b>2.0 SDK obsolète.....</b>	<b>8</b>
<b>Référence du paramètre.....</b>	<b>8</b>
Exemples.....	9
Envoi de votre premier message.....	9
Installation du module de noeud.....	9
Envoi de votre message.....	9
<b>Chapitre 5: Envoyer des messages en Python.....</b>	<b>10</b>
Syntaxe.....	10
Paramètres.....	10
Remarques.....	10
Exemples.....	10
Envoi de MMS.....	10
Envoi de SMS.....	11

Envoi de messages et MMS via Twilio.....	11
<b>Chapitre 6: Envoyer un message SMS en utilisant Bash et cURL.....</b>	<b>13</b>
Syntaxe.....	13
Paramètres.....	13
Exemples.....	13
Envoyer un seul message en utilisant un identifiant d'expéditeur alphanumérique.....	13
<b>Crédits.....</b>	<b>15</b>

---

# À propos

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](mailto:info@zzzprojects.com)

---

# Chapitre 1: Démarrer avec Twilio

## Remarques

Twilio fournit une plate-forme de communication via la messagerie IP (chat), vidéo, voix, SMS et MMS. Cet exemple simple montre comment envoyer un message texte SMS à l'aide du [module Twilio pour Node.js](#).

## Versions

Version	Date de sortie
1.0.0	2016-04-12

## Exemples

### Envoi d'un message SMS à partir d'un numéro américain

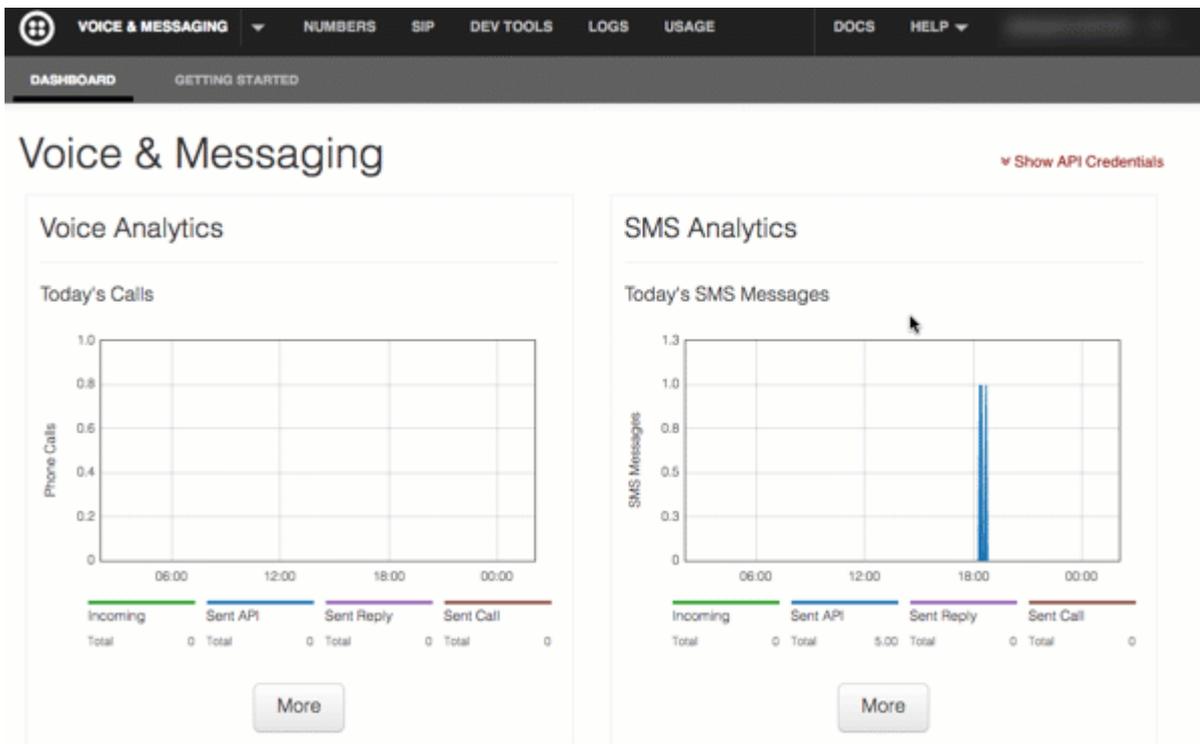
Voici comment envoyer un message texte SMS à partir d'un numéro américain à l'aide du SDK Node.js de Twilio.

Vous devez d'abord installer le client Node.js en utilisant:

```
npm install twilio
```

Ensuite, vous devez [créer un compte sur leur site Web](#) .

Une fois que vous avez un compte, vous avez besoin du SID de compte et du jeton d'authentification que vous pouvez trouver sur le tableau de bord en ligne.



Dans l'exemple de code ci-dessous, remplacez [Account SID] et [Auth Token] par ceux de votre compte.

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

## Envoi d'un message SMS à l'aide de promesses

L'API Node.JS de Twilio prend en charge de manière native les promesses, vous permettant d'utiliser des promesses lors de l'envoi de messages SMS (cet exemple a été pris et adapté directement à partir [des Docs 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);
  });
```

Lire Démarrer avec Twilio en ligne: <https://riptutorial.com/fr/twilio/topic/443/demarrer-avec-twilio>

---

# Chapitre 2: Comment envoyer un SMS avec Ruby

## Exemples

### Utilisation du Twilio Gem

Cela suppose que vous avez un compte twilio et que vous avez acheté / réservé un numéro de téléphone ...

Si vous utilisez bundler, ajoutez la ligne suivante à votre Gemfile pour inclure `twilio-ruby` dans votre projet:

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

sinon, entrez `gem install twilio-ruby` sur la ligne de commande. Vous pourriez avoir besoin de `sudo` si vous utilisez le système Ruby et non `ruby-env` etc.

et le code

```
# 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!")
```

Lire Comment envoyer un SMS avec Ruby en ligne:

<https://riptutorial.com/fr/twilio/topic/1760/comment-envoyer-un-sms-avec-ruby>

---

# Chapitre 3: Comment faire pour recevoir des messages SMS avec Ruby on Rails

## Exemples

### Utilisation du Twilio Gem

En supposant que vous avez un compte Twilio et des [informations d'identification API](#) , ajoutez ce qui suit à votre Gemfile:

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

Sinon, vous pouvez `gem install twilio-ruby` .

Pour que Twilio envoie un SMS entrant à un itinéraire particulier de votre application, vous devez configurer l'URL de messagerie pour votre [numéro de téléphone](#) . Une fois cela fait, vous devez configurer une route dans `config/routes.rb` :

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

Cela crée un ensemble de routes RESTful ( `GET /messages` pour lister les messages, `POST /messages` pour créer un message, etc.) qui enverront des requêtes au `MessagesController` . Une fois ces itinéraires définis, vous devez définir l'URL de messagerie du tableau de bord Twilio sur `https://your.site/messages` tant que Webhook HTTP POST.

Votre contrôleur gère le message:

```
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
```

```
end
```

Lire Comment faire pour recevoir des messages SMS avec Ruby on Rails en ligne:

<https://riptutorial.com/fr/twilio/topic/3673/comment-faire-pour-recevoir-des-messages-sms-avec-ruby-on-rails>

---

# Chapitre 4: Envoi de messages dans Node.js

## Introduction

Twilio vous permet d'envoyer et de recevoir des messages texte avec Node.js en utilisant la bibliothèque `twilio-node` pour passer des appels. Ce guide suppose que vous avez déjà configuré un compte avec Twilio et que vous disposez de votre compte SID et du jeton d'authentification de la console Twilio.

## Paramètres

Paramètre	Détails
<code>à</code>	Un numéro de téléphone valide pour recevoir le message
<code>de</code>	Un numéro Twilio qui vous est attribué
<code>corps</code>	Le corps du message texte limité à 1600 caractères
<code>StatusCallback</code>	Une URL vers laquelle Twilio publie un changement d'état du message
<code>maxPrix</code>	Définir le prix maximum d'un message
<code>période de validité</code>	Nombre de secondes pendant lesquelles le message restera dans la file d'attente Twilio
<code>fournir une réponse</code>	Valeur booléenne définie sur <code>true</code>
<code>mediaUrl</code>	Une URL contenant un contenu gif, png ou jpeg qui sera envoyé avec le message

## Remarques

---

### 2.0 SDK obsolète

Twilio a deux versions du module `twilio-node`, un SDK 2.0 et un SDK 3.0. Il est recommandé d'utiliser le SDK 3.0, car le SDK 2.0 sera obsolète le 31/08/2017.

Avis de dépréciation: les nouvelles fonctionnalités ne seront ajoutées qu'à la nouvelle bibliothèque (Node Helper Library 3.x). L'ancienne bibliothèque (2.x) sera officiellement prise en charge jusqu'au 31/08/2017. Après ce jour, Twilio cessera de fournir des correctifs de bogues et le support technique pourrait vous demander de mettre à niveau avant de déboguer les problèmes. <https://www.twilio.com/docs/libraries/node>

---

# Référence du paramètre

Vous pouvez vous référer aux [documents Twilio REST](#) pour une description plus détaillée. Twilio a également un [JSdocs](#) qui peut être utilisé comme référence.

## Exemples

### Envoi de votre premier message

## Installation du module de noeud

Vous pouvez installer ce module en exécutant la commande ci-dessous dans votre répertoire de projet:

```
npm install twilio
```

## Envoi de votre message

```
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));
```

Lire [Envoi de messages dans Node.js en ligne](#): <https://riptutorial.com/fr/twilio/topic/9955/envoi-de-messages-dans-node-js>

# Chapitre 5: Envoyer des messages en Python

## Syntaxe

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

## Paramètres

Paramètre	Détails
<code>à (str)</code>	Le numéro de téléphone de destination.
<code>de_ (str)</code>	Le numéro de téléphone envoyant ce message (doit être un numéro Twilio vérifié)
<code>corps (str)</code>	Le message que vous souhaitez envoyer, limité à 160 caractères.
<code>status_callback</code>	Une URL vers laquelle Twilio affichera son message lors du traitement.
<code>application_sid (str)</code>	Le caractère de 34 caractères de l'application que Twilio doit utiliser pour gérer cet appel téléphonique.

## Remarques

Twilio enverra des messages SMS pour les codes longs (aux États-Unis, un numéro normal à dix chiffres) à raison d'un message par seconde. Pour un débit supplémentaire, vous pouvez utiliser des pools de numéros avec [Copilot](#) ou demander un [code de fonction](#), qui peut envoyer 30 messages ou plus par seconde.

## Exemples

### Envoi de MMS

Vous pouvez ajouter une image à votre message en utilisant le paramètre `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'])
```

## Envoi de SMS

Pour envoyer vos premiers SMS avec Twilio et Python, il vous suffit de démarrer la bibliothèque d'assistance [Twilio-Python](#) .

```
# 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!")
```

## Envoi de messages et MMS via Twilio

**Twilio** aide à créer des applications qui communiquent avec tout le monde. Voice & Video, Messaging et Authentication API pour chaque application.

Vous pouvez obtenir une clé API gratuitement.

Pour envoyer un message via Twilio, votre application doit envoyer une requête HTTP POST à Twilio avec les éléments suivants:

1. Le numéro de téléphone auquel vous souhaitez envoyer le message.
2. Numéro Twilio à partir duquel vous essayez d'envoyer le message. (Seuls les numéros de téléphone Twilio activés fonctionnent)
3. Le corps du message.

---

Pour envoyer un SMS, faites une demande HTTP POST à la ressource Messages.

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

Vous trouverez ci-dessous un exemple de code pour montrer comment l'envoi de messages avec l'API Twilio fonctionnera.

```
# 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 le numéro Twilio prend en charge MMS, vous pouvez également envoyer et recevoir des MMS.

Vous trouverez ci-dessous le code pour envoyer des MMS via l'API Twilio.

```
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'
    ])
```

Lire Envoyer des messages en Python en ligne: <https://riptutorial.com/fr/twilio/topic/579/envoyer-des-messages-en-python>

# Chapitre 6: Envoyer un message SMS en utilisant Bash et cURL

## Syntaxe

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

## Paramètres

Paramètre	Détails
{Api version}	Date de version de l'API. par exemple 2010-04-01
{Compte Sid}	Le compte Sid. Commence ca
{corps}	Corps encodé en URL incluant De, À, Corps et MessagingServiceSid

## Exemples

### Envoyer un seul message en utilisant un identifiant d'expéditeur alphanumérique

Une demande HTTP POST est envoyée à une URL au format: " [https://api.twilio.com/2xxx-xx-xx/Accounts/\[AccountSid\]/Messages.json](https://api.twilio.com/2xxx-xx-xx/Accounts/[AccountSid]/Messages.json)

L'exemple ci-dessous utilise une chaîne alphanumérique comme expéditeur. Au moment de l'écriture, un identifiant d'expéditeur ne peut être ajouté que via une demande de service Twilio.

Exemple de demande:

```
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?}\" ;
```

## Exemple de réponse (JSON, formaté):

```
{
  "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"
  }
}
```

Lire Envoyer un message SMS en utilisant Bash et cURL en ligne:

<https://riptutorial.com/fr/twilio/topic/4237/envoyer-un-message-sms-en-utilisant-bash-et-curl>

# Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec Twilio	<a href="#">Adam</a> , <a href="#">cdock</a> , <a href="#">Community</a> , <a href="#">ddoria921</a> , <a href="#">Kevin Whinnery</a> , <a href="#">msmanek</a> , <a href="#">Sven</a>
2	Comment envoyer un SMS avec Ruby	<a href="#">derrek</a> , <a href="#">Steve Graham</a>
3	Comment faire pour recevoir des messages SMS avec Ruby on Rails	<a href="#">Benjamin Curtis</a>
4	Envoi de messages dans Node.js	<a href="#">Matt Shirley</a>
5	Envoyer des messages en Python	<a href="#">anthon</a> , <a href="#">Henk Langeveld</a> , <a href="#">Jacob Budin</a> , <a href="#">Megan Speir</a> , <a href="#">Shubham Gupta</a>
6	Envoyer un message SMS en utilisant Bash et cURL	<a href="#">antonycc</a>