



Intuit Inc. / IQX Corp.
Exhibit 1022
Page 1



VOICE

MESSAGING

CLIENT

IN ACTION

SHOWCASE

DOCS

HELP

LOGIN

twilio docs

Quickstart

HowTo's

Helper Libraries

API docs

Security

Get Started

Ruby

Twilio SMS and MMS Ruby Quickstart

Overview

Twilio Messaging allows you to:

1. Receive SMS and MMS messages to your Twilio phone numbers and reply back with SMS and MMS messages
2. Send SMS and MMS messages using Twilio's REST API
3. Track SMS conversations
4. Send SMS messages during phone calls

Receive and reply to SMS and MMS messages

When you buy a SMS-enabled Twilio phone number, you can associate that phone number with an SMS URL. When someone sends a text message to that phone number, Twilio makes an HTTP request to your URL with the body of the message and the sender's phone number. You can then respond to the SMS by returning a reply message in the HTTP response to Twilio.

Send SMS and MMS messages via the REST API

You can also send SMS and MMS messages using Twilio's **REST API**. To send a message, make an HTTP POST to Twilio with the body of the message and the phone number you want to send it to. SMS messages must be sent from Twilio SMS-enabled phone numbers due to the architecture of the global SMS network.

Track SMS conversations

Using standard HTTP cookies, you can share state across multiple messages allowing you to treat separate messages between any pair of phone numbers as a conversation, and store data about the conversation in the cookie for future reference.

Send SMS messages during phone calls

You can also use the TwiML `<Sms>` verb inside your existing Twilio voice calls to send a message during a phone conversation.

Quickstart Tutorials

Send and Receive SMS

1. [Receiving Incoming Text Messages](#)
2. [Replying to Incoming Messages](#)
3. [Sending SMS Messages](#)
4. [Tracking Conversations](#)
5. [Sending SMS via TwiML](#)

Handle Incoming Calls

1. [Say a Message](#)
2. [Greet Caller By Name](#)
3. [Play MP3 for Caller](#)
4. [Connect Call to Second Person](#)
5. [Record Caller's Voice](#)

Make Outgoing Calls

1. [Preparing a Call](#)
2. [Making an Outbound Call](#)
3. [Viewing Call Records](#)

Twilio Client for iOS

1. [Generating Capability Tokens](#)
2. [Make Outgoing Calls](#)
3. [Hanging Up Calls](#)
4. [Customize Your Application](#)
5. [Pass Data from the Device](#)
6. [Receive Incoming Calls](#)
7. [Call In and Out from the App](#)

Twilio Client for Android

1. [Generating Capability Tokens](#)
2. [Make Outgoing Calls](#)
3. [Hanging Up Calls](#)
4. [Customize Your Application](#)
5. [Pass Data from the Device](#)
6. [Receive Incoming Calls](#)

Frequently asked questions

For specific details on what you can and can't accomplish using Twilio SMS please refer to the [SMS FAQ](#)

Before you start

This tutorial assumes you have a Ruby development environment with [Sinatra](#) and the [twilio-ruby](#) helper library. If you don't have these yet, please see our post on [setting up your environment](#).

With that out of the way, let's go on to the next step.

[Next: Send SMS via the Twilio API »](#)

7. [Call In and Out from the App](#)

Twilio Client for JavaScript

1. [Hello Monkey Client](#)
2. [Hang Up Calls in the Browser](#)
3. [Receive Incoming Calls in the Browser](#)
4. [Make Outgoing Calls from the Browser](#)
5. [Browser-to-Browser Calls](#)
6. [Displaying Availability](#)

Twilio Connect

HowTo's and Examples

Helper Libraries

TwiML Reference

REST API Reference

SIP Reference

Client Reference

Connect Reference

Debugging Errors

Security

Availability & Reliability



[About Us](#) [Trust](#) [Blog](#) [Jobs](#) [Legal](#) [Help](#) [Talk to sales](#)

© 2009 - 2014 Twilio, Inc. All rights reserved.
Patents Pending. Twilio, TwiML, and OpenVBX are trademarks of Twilio, Inc.



VOICE

MESSAGING

CLIENT

IN ACTION

SHOWCASE

DOCS

HELP

LOGIN

twilio docs

Quickstart

HowTo's

Helper Libraries

API docs

Security

Get Started

Ruby

Twilio SMS and MMS SMS Quickstart

Sending Text Messages via the REST API

After responding to so many messages, I think it's time we threw a monkey party. We are going to invite our friends via SMS messages.

To send an outgoing SMS message, perform an HTTP POST to the **Messages resource URI**. Using the twilio-ruby library's `client` object, you can create a new instance of the `message` resource and specify the `to`, `from`, and `body` parameters for your message.

If you want to send a message to several recipients, you can create an array of recipients and iterate through each number in that array.

send-sms.rb

```

1  require 'rubygems'
2  require 'twilio-ruby'
3
4  account_sid = "ACxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
5  auth_token = "yyyyyyyyyyyyyyyyyyyyyyyyyyyy"
6  client = Twilio::REST::Client.new account_sid, auth_token
7
8  from = "+14159998888" # Your Twilio number
9
10
11 friends = {
12   "+14153334444" => "Curious George",
13   "+14155557775" => "Boots",
14   "+14155551234" => "Virgil"
15 }
16 friends.each do |key, value|
17   client.account.messages.create(
18     :from => from,
19     :to => key,
20     :body => "Hey #{value}, Monkey party at 6PM. Bring Bananas!"
21   )
22   puts "Sent message to #{value}"
23 end

```

Once your app is saved, run the file at the command line:

```

1  $ ruby send-sms.rb
2  Sent message to Curious George
3  Sent message to Boots
4  Sent message to Virgil

```

As soon as you hit enter, the app should send SMS messages to the specified recipients and you should see a confirmation message in your terminal.

Quickstart Tutorials

Send and Receive SMS

1. [Receiving Incoming Text Messages](#)
2. [Replying to Incoming Messages](#)
3. [Sending SMS Messages](#)
4. [Tracking Conversations](#)
5. [Sending SMS via TwiML](#)

Handle Incoming Calls

1. [Say a Message](#)
2. [Greet Caller By Name](#)
3. [Play MP3 for Caller](#)
4. [Connect Call to Second Person](#)
5. [Record Caller's Voice](#)

Make Outgoing Calls

1. [Preparing a Call](#)
2. [Making an Outbound Call](#)
3. [Viewing Call Records](#)

Twilio Client for iOS

1. [Generating Capability Tokens](#)
2. [Make Outgoing Calls](#)
3. [Hanging Up Calls](#)
4. [Customize Your Application](#)
5. [Pass Data from the Device](#)
6. [Receive Incoming Calls](#)
7. [Call In and Out from the App](#)

Twilio Client for Android

1. [Generating Capability Tokens](#)
2. [Make Outgoing Calls](#)
3. [Hanging Up Calls](#)
4. [Customize Your Application](#)
5. [Pass Data from the Device](#)
6. [Receive Incoming Calls](#)

Sending MMS Messages via the REST API

It's also easy to send an outgoing MMS using Twilio. To send an MMS, you also make an HTTP POST request to the [Messages resource](#) but this time specify a parameter for the URL of media, such as an image.

MMS messages can only be sent and received by numbers having MMS capability. You can [check the capabilities](#) of numbers in the account portal or query the [Available Phone Numbers](#) resource to search for Twilio numbers that are MMS enabled.

Modify send-sms.rb as follows:

send-sms.rb

```
1 | friends.each do |key, value|
2 |   client.account.messages.create(
3 |     :from => from,
4 |     :to => key,
5 |     :body => "Hey #{value}, Monkey party at 6PM. Bring Bananas!"
6 |     :media_url => "https://demo.twilio.com/owl.png"
7 |   )
8 |   puts "Sent message to #{value}"
9 | end
```

For more information about sending SMS and MMS messages, see the [REST documentation](#).

[Next: Reply to Incoming SMS »](#)

[7. Call In and Out from the App](#)

Twilio Client for JavaScript

- [1. Hello Monkey Client](#)
- [2. Hang Up Calls in the Browser](#)
- [3. Receive Incoming Calls in the Browser](#)
- [4. Make Outgoing Calls from the Browser](#)
- [5. Browser-to-Browser Calls](#)
- [6. Displaying Availability](#)

Twilio Connect

HowTo's and Examples

Helper Libraries

Twiml Reference

REST API Reference

SIP Reference

Client Reference

Connect Reference

Debugging Errors

Security

Availability & Reliability



[About Us](#) [Trust](#) [Blog](#) [Jobs](#) [Legal](#) [Help](#) [Talk to sales](#)

© 2009 - 2014 Twilio, Inc. All rights reserved.
Patents Pending. Twilio, TwiML, and OpenVBX are trademarks of Twilio, Inc.



VOICE

MESSAGING

CLIENT

IN ACTION

SHOWCASE

DOCS

HELP

LOGIN

twilio docs

Quickstart

HowTo's

Helper Libraries

API docs

Security

Get Started

Ruby

Twilio SMS and MMS Ruby Quickstart

Respond to an Incoming SMS

We'll start with replying to SMS messages. When someone sends a text message to your Twilio number, you can send a reply back. Here's how:

This tutorial assumes you have a Ruby development environment with [Sinatra](#) and the [twilio-ruby](#) helper library. If you don't have these yet, please see our [post on setting up your environment](#).

Create a new Sinatra application with the following:

sms-quickstart.rb

```
1 require 'rubygems'
2 require 'twilio-ruby'
3 require 'sinatra'
4
5
6 get '/sms-quickstart' do
7   twiml = Twilio::TwiML::Response.new do |r|
8     r.Message "Hey Monkey. Thanks for the message!"
9   end
10  twiml.text
11 end
```

We're using the [twilio-ruby](#) helper library to generate TwiML, Twilio's Markup XML.

Save the file and start your local server at the command line:

```
1 $ ruby sms-quickstart.rb
2 >> Sinatra has taken the stage ...
3 >> Listening on 0.0.0.0:4567
```

Go to your browser at <http://localhost:4567/sms-quickstart>. You should see TwiML returned that looks like this:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <Response>
3   <Message>Hey Monkey. Thanks for the message!</Message>
4 </Response>
```

If so, great. You're ready to move on to the next step.

Quickstart Tutorials

Send and Receive SMS

1. [Receiving Incoming Text Messages](#)
2. [Replying to Incoming Messages](#)
3. [Sending SMS Messages](#)
4. [Tracking Conversations](#)
5. [Sending SMS via TwiML](#)

Handle Incoming Calls

1. [Say a Message](#)
2. [Greet Caller By Name](#)
3. [Play MP3 for Caller](#)
4. [Connect Call to Second Person](#)
5. [Record Caller's Voice](#)

Make Outgoing Calls

1. [Preparing a Call](#)
2. [Making an Outbound Call](#)
3. [Viewing Call Records](#)

Twilio Client for iOS

1. [Generating Capability Tokens](#)
2. [Make Outgoing Calls](#)
3. [Hanging Up Calls](#)
4. [Customize Your Application](#)
5. [Pass Data from the Device](#)
6. [Receive Incoming Calls](#)
7. [Call In and Out from the App](#)

Twilio Client for Android

1. [Generating Capability Tokens](#)
2. [Make Outgoing Calls](#)
3. [Hanging Up Calls](#)
4. [Customize Your Application](#)
5. [Pass Data from the Device](#)
6. [Receive Incoming Calls](#)

Let Twilio Find Your Server

At this point in the tutorial, you will need to find a way to expose your server to the public Internet so Twilio can reach it.

Here are some tutorials that may be able to help you:

- [Ngrok \(for testing your local server\)](#)
- [Getting Started with Ruby on Heroku](#)
- [Sample Sinatra app on Dotcloud](#)
- [Deploying Sinatra behind Nginx and Unicorn](#)

For the rest of the tutorial, we will assume your server is available at example.com.

Copy and paste the URL of your server into the "SMS" URL of a number on the [Numbers](#) page of your Twilio Account. On the page for that number, change the Method from "POST" to "GET".

Try It Out

Now, send a text message to your Twilio number. You should get a message back, "Hey Monkey. Thanks for the message!"

Respond with an MMS

Now modify your send-quickstart.rb on your server as follows specifying a URL of an image. If necessary, restart your server, then text your Twilio number again. You should receive a text message in response saying, "Hello, Mobile Monkey", but this time images will be included!

MMS messages can only be sent and received by numbers having MMS capability. You can [check the capabilities](#) of numbers in the account portal or query the [Available Phone Numbers](#) resource to search for Twilio numbers that are MMS enabled.

sms-quickstart.rb

```
1
2 require 'rubygems'
3 require 'twilio-ruby'
4 require 'sinatra'
5
6 get '/sms-quickstart' do
7   twiml = Twilio::Twiml::Response.new do |r|
8     r.Message do |message|
9       message.Body "Body"
10      message.MediaUrl "https://demo.twilio.com/owl.png"
11      message.MediaUrl "https://demo.twilio.com/logo.png"
12    end
13  end
14  twiml.text
15 end
```

[Next: Replying to Messages »](#)

7. Call In and Out from the App

Twilio Client for JavaScript

1. [Hello Monkey Client](#)
2. [Hang Up Calls in the Browser](#)
3. [Receive Incoming Calls in the Browser](#)
4. [Make Outgoing Calls from the Browser](#)
5. [Browser-to-Browser Calls](#)
6. [Displaying Availability](#)

Twilio Connect

HowTo's and Examples

Helper Libraries

Twiml Reference

REST API Reference

SIP Reference

Client Reference

Connect Reference

Debugging Errors

Security

Availability & Reliability



[About Us](#) [Trust](#) [Blog](#) [Jobs](#) [Legal](#) [Help](#) [Talk to sales](#)

© 2009 - 2014 Twilio, Inc. All rights reserved.
Patents Pending. Twilio, TwiML, and OpenVBX are
trademarks of Twilio, Inc.



PRODUCTS ▾

PRICING ▾

SOLUTIONS ▾

API & DOCS ▾

HELP

SIGNUP

LOGIN

TWILIO DOCS

Quickstart

How-Tos

Helper Libraries

API Docs

Security

Twilio SMS and MMS Ruby Quickstart



Replying to Incoming Messages

OK, we've said "Hello" in reply to people who have sent SMS messages to us, but what if we want to customize our response in some way?

For example, we're going to make an app that greets the sender by name if they're in our contacts list and send back a general greeting if they're not:

sms-quickstart.rb

```

1 require 'rubygems'
2 require 'twilio-ruby'
3 require 'sinatra'
4
5
6 get '/sms-quickstart' do
7   sender = params[:From]
8   friends = {
9     "+14153334444" => "Curious George",
10    "+14158157775" => "Boots",
11    "+14155551234" => "Virgil"
12  }
13   name = friends[sender] || "Mobile Monkey"
14   twiml = Twilio::TwiML::Response.new do |r|
15     r.Message "Hello, #{name}. Thanks for the message."
16   end
17   twiml.text
18 end

```

Save your own (non-Twilio) phone number and name in the `friends` array above. Then, do the following steps to make your app live:

1) Open the terminal and start your server with Sinatra:

```

1 $ ruby sms-quickstart.rb
2 >> Sinatra has taken the stage ...
3 >> Listening on 0.0.0.0:4567

```

2) Go to <http://localhost:4567/sms-quickstart> and confirm you get TwiML back.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <Response>
3   <Message>Hello, Mobile Monkey. Thanks for the message.</Message>
4 </Response>

```

Now, try sending your Twilio number a text. If you add your phone number and name to the `friends` array you will get a personalized SMS thank-you message from your app.

Quickstart Tutorials

Send and Receive SMS

1. [Receiving Incoming Text Messages](#)
2. [Replying to Incoming Messages](#)
3. [Sending SMS Messages](#)
4. [Tracking Conversations](#)
5. [Sending SMS via TwiML](#)

Handle Incoming Calls

1. [Say a Message](#)
2. [Greet Caller By Name](#)
3. [Play MP3 for Caller](#)
4. [Connect Call to Second Person](#)
5. [Record Caller's Voice](#)

Make Outgoing Calls

1. [Preparing a Call](#)
2. [Making an Outbound Call](#)
3. [Viewing Call Records](#)

Twilio Client for iOS

1. [Setting up your Server](#)
2. [Making Outgoing Calls](#)
3. [Hanging up from your iOS Device](#)
4. [Connecting to a TwiML Application](#)
5. [Passing Parameters to your Application](#)
6. [Receiving Incoming Calls](#)
7. [Making Calls in/out of your App](#)

Twilio Client for Android

1. [Setting up your Server](#)
2. [Making Outgoing Calls](#)
3. [Hanging up from your Android Device](#)
4. [Connecting to a TwiML Application](#)

Next: Tracking Conversations »

5. [Passing Parameters to your Application](#)
6. [Receiving Incoming Calls](#)
7. [Making Calls in/out of your App](#)

Twilio Client for JavaScript

1. [Hello Monkey Client](#)
2. [Hang Up Calls in the Browser](#)
3. [Receive Incoming Calls in the Browser](#)
4. [Make Outgoing Calls from the Browser](#)
5. [Browser-to-Browser Calls](#)
6. [Displaying Availability](#)

Twilio Connect

- [How-Tos and Examples](#)
- [Helper Libraries](#)
- [Twiml Reference](#)
- [REST API Reference](#)
- [SIP Reference](#)
- [Elastic SIP Trunking Reference](#)
- [Network Traversal Service Reference](#)
- [Client Reference](#)
- [Connect Reference](#)
- [Debugging Errors](#)
- [Security](#)
- [Availability & Reliability](#)
- [Learning Resources](#)

COMPANY

[About Us](#)
[Team](#)
[Jobs](#)

PRODUCTS

[Voice](#)
[Client](#)
[Messaging](#)

RESOURCES

[API & Docs](#)
[Security](#)
[International](#)
[Open Source](#)

COMMUNITY

[Twilio Blog](#)
[Engineering Blog](#)
[Twilio.org](#)

SOLUTIONS

[Elements](#)
[Showcase](#)
[Partners](#)
[DOers](#)

SUPPORT

[Help Center](#)
[Status](#)
[Talk to Sales](#)

© 2009 - 2015 Twilio, Inc. All rights reserved. Patents Pending.



[Terms of Service](#) | [Privacy Policy](#)



VOICE

MESSAGING

CLIENT

IN ACTION

SHOWCASE

DOCS

HELP

LOGIN

twilio docs

Quickstart

HowTo's

Helper Libraries

API docs

Security

Get Started

Ruby

Twilio SMS and MMS Ruby Quickstart

Tracking SMS Conversations

Twilio offers a simple way to keep track of conversations between two phone numbers by using HTTP cookies.

What is a cookie?

Just like in web applications, a cookie is a small file that your application can store on Twilio's servers to keep track of information, such as a username or account. For Twilio SMS, cookies are scoped to the "conversation" between two parties -- you can have a unique cookie for each To/From phone number pair. For example, you can store a unique cookie for any messages sent between 415-555-2222 and 415-555-1111, which will be different than the cookie used between 415-555-3333 and 415-555-1111.

Why cookies?

In web apps, you write a cookie to keep "statefulness" between separate requests from the same browser. Similarly, SMS messages are independent communications between two parties, so Twilio allows you to tie them together as a logical session via cookies. This means you can use server-side sessions to keep track of application state between requests. How cool! Twilio will expire the cookies for that conversation after four hours of inactivity, as if the user "closed the browser."

Storing Conversation Data

The cookies let you share state across multiple messages allowing you to treat separate messages as a conversation, and store data about the conversation in the cookies for future reference.

Lets try using session counters to see if a particular user has messaged us before. If they're a new sender, we'll thank them for the new message. If they've sent us messages before, we'll specify how many messages we've gotten from them.

sms-quickstart.rb

```

1
2 require 'rubygems'
3 require 'twilio-ruby'
4 require 'sinatra'
5
6 get '/sms-quickstart' do
7   session["counter"] ||= 0

```

Quickstart Tutorials

Send and Receive SMS

1. [Receiving Incoming Text Messages](#)
2. [Replying to Incoming Messages](#)
3. [Sending SMS Messages](#)
4. [Tracking Conversations](#)
5. [Sending SMS via TwiML](#)

Handle Incoming Calls

1. [Say a Message](#)
2. [Greet Caller By Name](#)
3. [Play MP3 for Caller](#)
4. [Connect Call to Second Person](#)
5. [Record Caller's Voice](#)

Make Outgoing Calls

1. [Preparing a Call](#)
2. [Making an Outbound Call](#)
3. [Viewing Call Records](#)

Twilio Client for iOS

1. [Generating Capability Tokens](#)
2. [Make Outgoing Calls](#)
3. [Hanging Up Calls](#)
4. [Customize Your Application](#)
5. [Pass Data from the Device](#)
6. [Receive Incoming Calls](#)
7. [Call In and Out from the App](#)

Twilio Client for Android

1. [Generating Capability Tokens](#)
2. [Make Outgoing Calls](#)
3. [Hanging Up Calls](#)
4. [Customize Your Application](#)
5. [Pass Data from the Device](#)
6. [Receive Incoming Calls](#)

```

8 | sms_count = session["counter"]
9 | if sms_count == 0
10 |   message = "Hello, thanks for the new message."
11 | else
12 |   message = "Hello, thanks for message number #{sms_count + 1}"
13 | end
14 | twiml = Twilio::Twiml::Response.new do |r|
15 |   r.Sms message
16 | end
17 | session["counter"] += 1
18 | twiml.text
   | end

```

Go ahead and send a few messages to your Twilio application, and watch your counter grow.

Next: Sending SMS Messages During a Phone Call »

7. **Call In and Out from the App**

Twilio Client for JavaScript

1. **Hello Monkey Client**
2. **Hang Up Calls in the Browser**
3. **Receive Incoming Calls in the Browser**
4. **Make Outgoing Calls from the Browser**
5. **Browser-to-Browser Calls**
6. **Displaying Availability**

Twilio Connect

HowTo's and Examples

Helper Libraries

Twiml Reference

REST API Reference

SIP Reference

Client Reference

Connect Reference

Debugging Errors

Security

Availability & Reliability



[About Us](#) [Trust](#) [Blog](#) [Jobs](#) [Legal](#) [Help](#) [Talk to sales](#)

© 2009 - 2014 Twilio, Inc. All rights reserved.
Patents Pending. Twilio, Twiml, and OpenVBX are trademarks of Twilio, Inc.



VOICE

MESSAGING

CLIENT

IN ACTION

SHOWCASE

DOCS

HELP

LOGIN

twilio docs

Quickstart

HowTo's

Helper Libraries

API docs

Security

Get Started

Ruby

Twilio SMS and MMS Ruby Quickstart

Sending Text Messages from a Phone Call

Did you know you can also send SMS when someone calls your Twilio number? The `<Sms>` verb is actually a valid response to an incoming call. Give this a try:

```

1 |
2 | require 'rubygems'
3 | require 'twilio-ruby'
4 | require 'sinatra'
5 |
6 | get '/sms-quickstart' do
7 |   Twilio::Twiml::Response.new do |r|
8 |     r.Say "Hello Monkey! You will get an SMS message soon."
9 |     r.Sms "Here it is!"
10 |   end.text
    end

```

If you go that that app in your browser, you should see TwiML that looks like this:

```

1 |
2 | <?xml version="1.0" encoding="UTF-8"?>
3 | <Response>
4 |   <Say>Hello Monkey! You will get an SMS message soon.</Say>
5 |   <Sms>Here it is!</Sms>
    </Response>

```

Save a publicly-accessible URL to return that XML. Then, configure that URL to be the Voice URL of your Twilio phone number.

Call that number from a mobile phone -- you should hear a message *and* get a text message in response.

And that's all she wrote folks! [Back to Quickstarts](#).

Quickstart Tutorials

Send and Receive SMS

1. [Receiving Incoming Text Messages](#)
2. [Replying to Incoming Messages](#)
3. [Sending SMS Messages](#)
4. [Tracking Conversations](#)
5. [Sending SMS via TwiML](#)

Handle Incoming Calls

1. [Say a Message](#)
2. [Greet Caller By Name](#)
3. [Play MP3 for Caller](#)
4. [Connect Call to Second Person](#)
5. [Record Caller's Voice](#)

Make Outgoing Calls

1. [Preparing a Call](#)
2. [Making an Outbound Call](#)
3. [Viewing Call Records](#)

Twilio Client for iOS

1. [Generating Capability Tokens](#)
2. [Make Outgoing Calls](#)
3. [Hanging Up Calls](#)
4. [Customize Your Application](#)
5. [Pass Data from the Device](#)
6. [Receive Incoming Calls](#)
7. [Call In and Out from the App](#)

Twilio Client for Android

1. [Generating Capability Tokens](#)
2. [Make Outgoing Calls](#)
3. [Hanging Up Calls](#)
4. [Customize Your Application](#)
5. [Pass Data from the Device](#)
6. [Receive Incoming Calls](#)

7. [Call In and Out from the App](#)

Twilio Client for JavaScript

1. [Hello Monkey Client](#)
2. [Hang Up Calls in the Browser](#)
3. [Receive Incoming Calls in the Browser](#)
4. [Make Outgoing Calls from the Browser](#)
5. [Browser-to-Browser Calls](#)
6. [Displaying Availability](#)

Twilio Connect

HowTo's and Examples

Helper Libraries

Twiml Reference

REST API Reference

SIP Reference

Client Reference

Connect Reference

Debugging Errors

Security

Availability & Reliability



[About Us](#) [Trust](#) [Blog](#) [Jobs](#) [Legal](#) [Help](#) [Talk to sales](#)

© 2009 - 2014 Twilio, Inc. All rights reserved.
Patents Pending. Twilio, Twiml, and OpenVBX are
trademarks of Twilio, Inc.