# Whatsapp client

## 1 Introduction

You can use the following guidelines if you want to use WhatsApp as client for your project. **Please note that Twilio doesn't support group chatbots yet!**

### 1.1 WhatsApp API

You have to note that the current WhatsApp API is still in Beta version, it was only released recently. It is very complex to use and requires authorizations from facebook before being run. Despite that we can already experiment with it using Twilio, which offers a much more comfortable API to work with.

## 2 Twilio

We will use Twilio (https://www.twilio.com/) for this, which offers a free tier to experiment with the API. First create an account before proceeding with your project's creation.

### 2.1 Creating a project

1. Start by creating a new project, selecting products and **SMS programmable API**
2. After giving it a name, you can invite your teammates to work together on your project
3. You now have access to your project's dashboard, on the left you have a menu with 3 icons, click the second one (Programmable SMS)
4. Click WhatsApp (Beta), a modal will open asking you to start your sandbox. That's how Twilio can make this API work, it will provide you with a "fake" phone number which will be your bot. In whatsApp, all bots must have a phone number. Twilio takes care of that for you.

### 2.2 Linking your project to your phone

You're now in the linking setup wizard.

1. First, you'll see a phone number displayed, start by sending it a message in whatsapp with the given code as message content. This will allow Twilio to enable the fake phone number and verify your phone (for me it was *join machinery-as*.
2. If everything worked well Twilio will answer your message. You can then click on the next button.
3. Here you'll have the possibility to send a template message, we'll let you read on what it is about in the documentation. Click next.
4. Next you'll be asked to send a two-way message, which means sending a message from your phone and having the bot answer it. Once your message is sent you'll see its details on your screen. Click next to configure your sandbox.

### 2.3 Configuring your sandbox

Your sandbox simply requires two URLs to be set, which are:

1. When a message comes in, which is required, triggered when a message is sent to your bot and which will receive the message as well as information about who sent it
2. status callback URL which is optional and allows you to receive status information

### 2.4 Public IP address

In order to map a public IP address to a localhost IP address, you have multiple possibilities.

1. Upload your code on heroku (or other) to have a public IP set, but requires to re-upload it whenever you wish to change your code
2. Use a software to map a public IP to your localhost address. Personnaly I use ngrok (https://ngrok.com/) for this. It offers a free tier, the only problem is that whenever I start the process it gives me a new public address, I then have to go change it in my Twilio sandbox, but it's a 1 minute process.

### 2.5 Using Ngrok

Once you have downloaded Ngrok, you can launch it with the following command: *ngrok http 8080* which will map a plugin IP address to your *localhost:8080*. You can then see that it exposes both an HTTP as well as an HTTPS for you. You can then copy the public address (https://randomstuff.ngrok.io) to your sandbox.

## 3 Coding

Once you're done with all this setup you can actually start coding! In this repository you'll find a basic Node application with express which answers messages very simply, you can use it to get started quickly.

### 3.1 Retrieving your tokens

To work with Twilio you'll need two tokens:

1. Your account SID
2. your auth token

Both of them can be found in your console. You'll need to provide it to Twilio's SDK to work with their API.