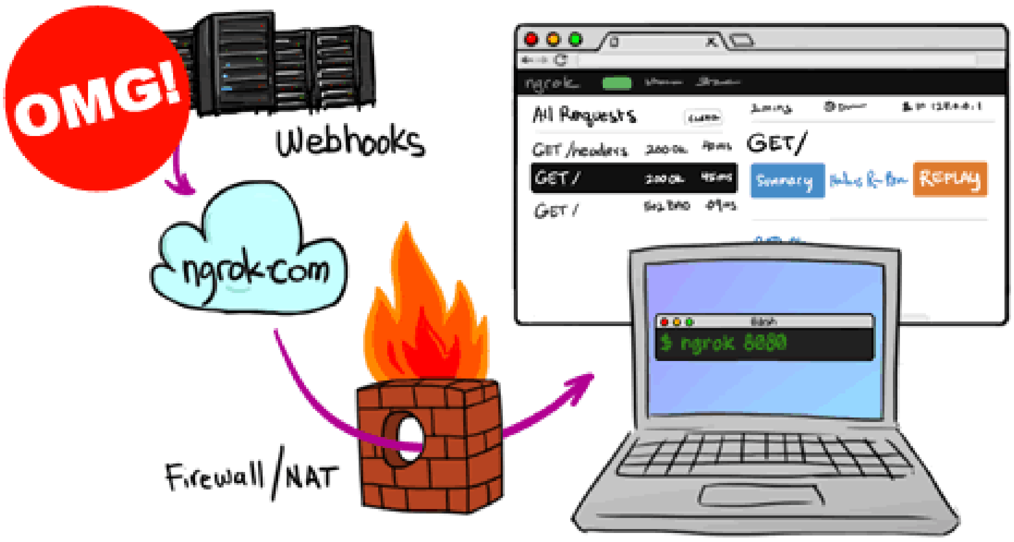
6 awesome reasons to use ngrok when testing webhooks



by Phil Nash (https://www.twilio.com/blog/author/phil) on September 10, 2015 (https://www.twilio.com/blog/2015/09/6-awesome-reasons-to-use-ngrok-when-testing-webhooks.html)

Like Tweet Follow @twilio

Webhooks are great! They are a perfect way for API services like Twilio to notify your application of events such as incoming calls or text messages. The one difficulty is that when you are developing an application that consumes webhooks you need a publicly accessible URL to configure the webhook service with. Typically you would develop on localhost and the rest of the world would have no access to your application, so how do you test your webhooks?

Well, that's where ngrok (http://ngrok.com) comes in. ngrok is a free tool that allows us to tunnel from a public URL to our application running locally. Here's my 6 favourite things about ngrok:

1. It's really easy to install and use

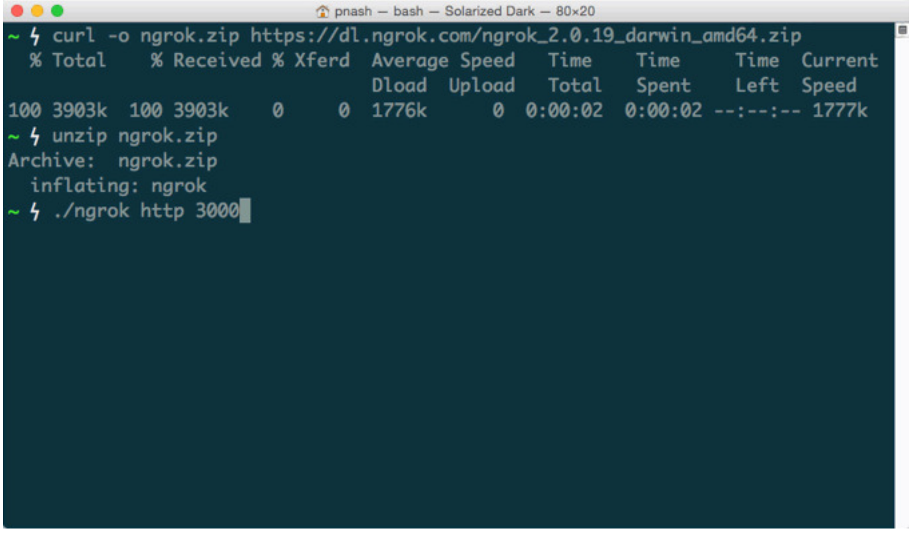
ngrok is built in Go so it is packaged as binaries for each major platform. To install ngrok do the following:

1. Download the package for your system
2. Unzip the package
3. There is no step 3!

Well, there is no step 3 if you have a Mac or Linux, if you're on Windows you might have a bit more work to do. Thankfully Devin's got you covered with his post on how to configure ngrok on Windows (https://www.twilio.com/blog/2014/03/configure-windows-for-local-webhook-testing-using-ngrok.html).

Once you have ngrok installed, using it to tunnel to an application running on, say, port 3000 is as easy as:

```shell
$ ./ngrok http 3000
```

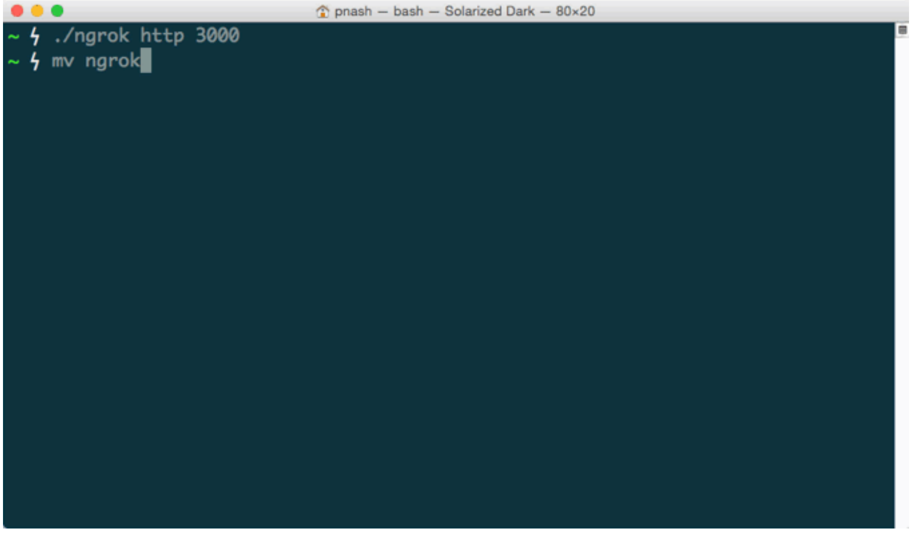


```
~ ϟ curl -o ngrok.zip https://dl.ngrok.com/ngrok_2.0.19_darwin_amd64.zip
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 3903k  100 3903k    0     0  1776k      0  0:00:02  0:00:02 --:--:-- 1777k
~ ϟ unzip ngrok.zip
Archive:  ngrok.zip
  inflating: ngrok
~ ϟ ./ngrok http 3000
```

Bonus install tip!

Once you've unzipped the ngrok executable move it to a folder that's in your `$PATH` then you can run ngrok from anywhere you want on the command line by just typing:

```Shell
$ ngrok http 3000
```



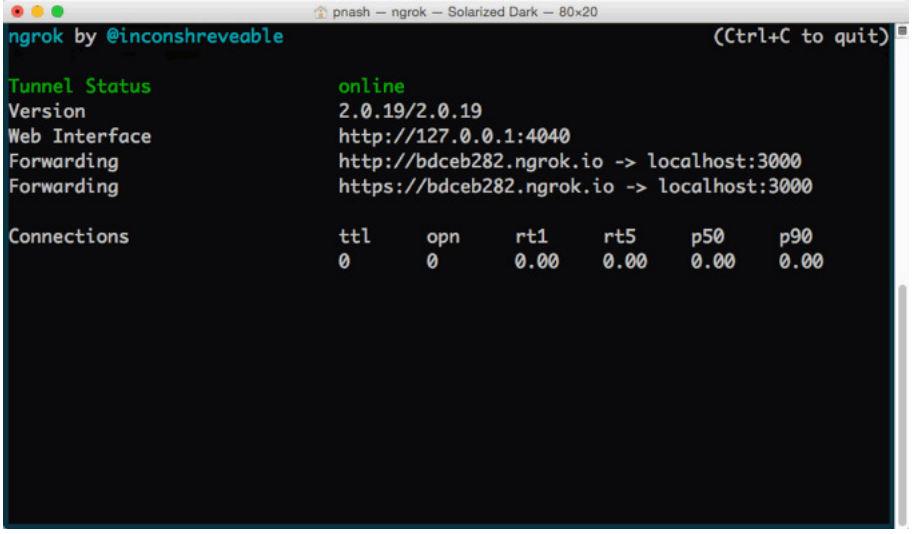
```
~ ϟ ./ngrok http 3000
~ ϟ mv ngrok
```

If you want a more in depth article on how to hook up ngrok with your Twilio apps, check out Kevin's post on getting started with ngrok (https://www.twilio.com/blog/2013/10/test-your-webhooks-locally-with-ngrok.html). Watch out though, as the commands have changed with the release of ngrok 2. You can follow the post, but substitute the syntax above to make sure it works.

2. It has a dashboard

When you start up ngrok it shows you the URL you are tunnelling through and another local URL, http://127.0.0.1:4040 (http://127.0.0.1:4040). Open that up and gaze in wonder at the dashboard in front of you.

The dashboard allows you to see the status of your tunnel, the requests you have made through it and the responses that came back from your application. Great for debugging misfiring webhook endpoints.



```
ngrok by @inconshreveable                                       (Ctrl+C to quit)

Tunnel Status                 online
Version                       2.0.19/2.0.19
Web Interface                 http://127.0.0.1:4040
Forwarding                    http://bdceb282.ngrok.io -> localhost:3000
Forwarding                    https://bdceb282.ngrok.io -> localhost:3000

Connections                   ttl     opn     rt1     rt5     p50     p90
                              0       0       0.00    0.00    0.00    0.00
```

3. You can replay requests

Take a look again at that dashboard. Not only can you inspect requests that came through the ngrok tunnel, you can replay them.

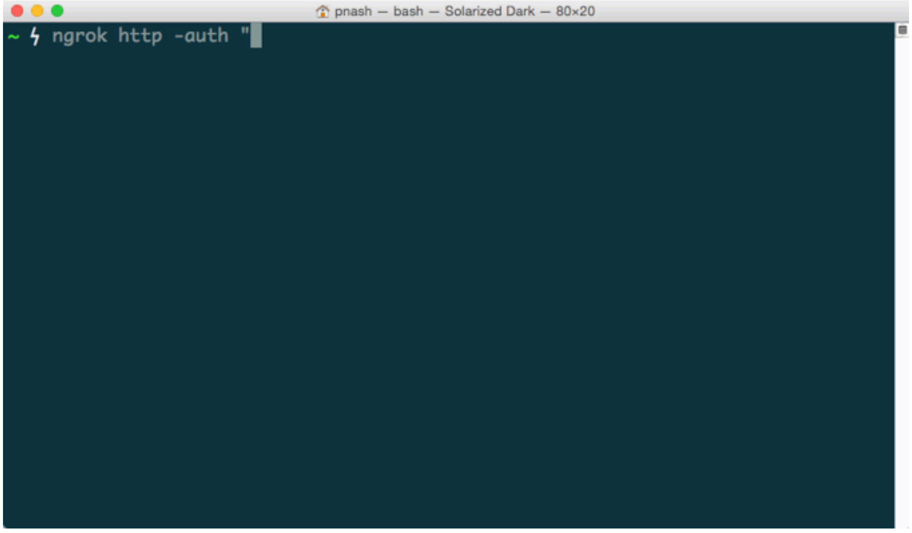
Now when your SMS endpoint is misbehaving, rather than sending another message and then another message until you get it right, you can just replay the incoming request from the dashboard.

4. You can secure your tunnels

Worried about someone snooping on the super secret app you're developing but still need to open a tunnel to check something? Stop worrying right now and check this out!

```Shell
$ ngrok http -auth "user:password" 3000
```

Boom! Your tunnel is now protected using HTTP auth. Noone's spying on your business!



```
~ ϟ ngrok http -auth "
```

Pro tips!

We've got into the pro tips section. Why are these "pro" tips? Because you need an ngrok pro account to use them! ngrok is always free to use, but to make sure it's sustainable, some sweet features have been held back for paying customers.

Check out the available plans (https://ngrok.com/product#pricing) and then check out my favourite feature below.

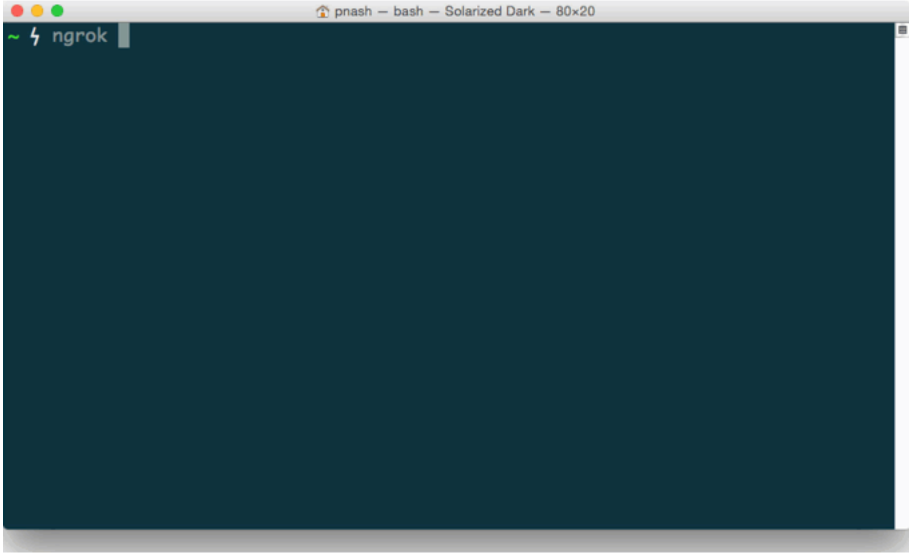
5. [Pro tip] You can use custom subdomains

Getting a new random subdomain every time you power up ngrok can be frustrating. Reconfiguring a webhook service every time you want to test is a total pain.

As a paying customer, ngrok allows you to use a custom subdomain, simply by providing it in the command. If you have a paid ngrok account try this out:

```shell
$ ngrok http -subdomain=thisisreallycool 3000
```

Take a look at the ngrok dashboard and hold on to your hat, you're now tunneling through the domain http://thisisreallycool.ngrok.io (http://thisisreallycool.ngrok.io).



```
~ ϟ ngrok
```

6. [Pro tip] You can configure all this to happen automatically

Now you know and love ngrok subdomain but you're looking at that command thinking, "That's a lot of typing every time and I've got to remember the arguments!"

Do not worry, this next feature is for you.

ngrok can actually read a bunch of useful stuff from a configuration file. If you have created a paid account and added your authtoken then you will have the config file `.ngrok2/ngrok.yml` in your `$HOME` directory.

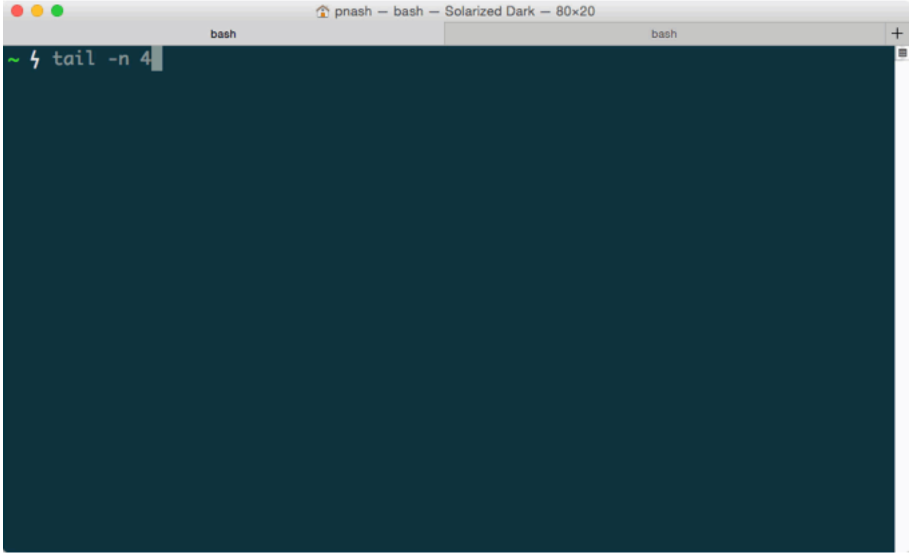
Now, we can recreate the subdomain config under a named tunnel, open up `~/.ngrok2/ngrok.yml` and paste in the following under the authtoken:

```
tunnels:
  cool:
    proto: http
    addr: 3000
    subdomain: thisisreallycool
```

Now you can run:

```Shell
$ ngrok start cool
```

and your tunnel will start up with the options you supplied in the yml file.



```
~ ϟ tail -n 4
```

ngrok is the best!

Now you know the power of ngrok testing your webhooks will never be the same again. If you want to learn more about ngrok you can find more features in the ngrok documentation (https://ngrok.com/docs). And why not check out this video of Alan Shreve (https://twitter.com/inconshreveable), creator of ngrok, speaking about the making of ngrok (https://www.twilio.com/signal/2015/videos/the-making-of-ngrok) at Twilio's Signal Conference (https://www.twilio.com/signal/2015) earlier this year.

If you have any other ngrok tips I'd love it if you would share them with me at @philnash (https://twitter.com/philnash). Bye for now!

by Phil Nash (https://www.twilio.com/blog/author/phil) pnash@twilio.com (mailto:pnash@twilio.com) @philnash (http://twitter.com/philnash)

Use Twilio Tutorials to Start Building

Call Tracking

(https://www.twilio.com/docs/tutorials/walkthrough/call-tracking/php/laravel)

SMS and MMS Notifications

(https://www.twilio.com/docs/tutorials/walkthrough/server-notifications/php/laravel)

Two-Factor Authentication with Authy

(https://www.twilio.com/docs/tutorials/walkthrough/two-factor-authentication/php/laravel)

6 Comments **Twilio Blog** 1 **Login**

Recommend 1 Share Sort by Oldest

Join the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

Mohamed Amro • 2 years ago

its great , but i would better if we can install it on DVR (used for Cam Recording) which is Linux based to open the ports to see Camera from aboard .. any Tutorial .

Reply • Share ›

philnash Mohamed Amro • 2 years ago

I'm afraid I've not heard of DVR before so I don't know of any tutorials to use ngrok with it. I will let you know if I come across anything though!

1 • Reply • Share ›

Tasumin Mohamed Amro • 2 years ago

Get a raspberry pi and run ngrok on it.
setup it as a cron to rerun at boot and place in /etc/rc.d
get the ip of the dvr and port ngrok to that ip/port

example:

dvr is 192.168.1.100
port is 8080

run this

ngrok 192.168.1.100:8080

Reply • Share ›

philnash Tasumin • 2 years ago

Glad you discovered how to manage this. Thanks for sharing!

Reply • Share ›

Pierre • a year ago

I tried to setup the auth but could not get it to work - might have been due to existing auth in webhook I was trying to reach (GitHub). Fun otherwise!

Reply • Share ›

philnash Pierre • a year ago

It is a lot of fun! If you are still having difficulties, feel free to drop me an email at philnash@twilio.com and I'll try my best to help you out.

Reply • Share ›

Subscribe Add Disqus to your siteAdd DisqusAdd Privacy

Search …

Power modern communications. Build the next generation of voice and SMS applications.