

# Forward a Call via Voice Proxy with PHP

Last updated on May 13, 2021

Today's post shows you how to proxy a voice call so that it appears to come from another number.

This is not as dubious as it sounds: there are many compelling business reasons why you might want to hide a caller's real number from other parties in the call.

A classic example is an online taxi service. To make the booking experience as smooth as possible, you want your customer and driver to be able to communicate with each other. But you don't want the driver to know the customer's real number (because you want to protect that customer's privacy) and, conversely, you don't want your customer to know your driver's number and book rides directly without using your service.

Luckily, this is really easy to do in PHP with the Nexmo Voice API, so let's get to it! The [complete source code](#) is available on Github.

Steps:

1. Install dependencies
2. Define the webhook endpoints

3. Log call events
4. Make your webhooks accessible
5. Purchase a number
6. Create a Nexmo Voice API application
7. Link the Application to your Nexmo number
8. Call your Nexmo virtual number
9. Implement the proxy logic
10. Test the voice proxy

## Vonage API Account

To complete this tutorial, you will need a [Vonage API account](#). If you don't have one already, you can [sign up today](#) and start building with free credit. Once you have an account, you can find your API Key and API Secret at the top of the [Vonage API Dashboard](#).

This tutorial also uses a virtual phone number. To purchase one, go to *Numbers > Buy Numbers* and search for one that meets your needs.

## Install Dependencies

To work with inbound calls using Nexmo's Voice API, you

must provide [webhooks](#) so that Nexmo can send data to your application.

We'll use `slim` to code these webhooks: a lightweight web framework for PHP. You can find out more about `slim` [here](#).

Install `slim` using composer:

```
composer require slim/slim:^3.8
```

---

## Define the Webhook Endpoints

Create a file called `index.php` that contains the following code:

```
<?php
use \Psr\Http\Message\ResponseInterface as Response;
use \Psr\Http\Message\ServerRequestInterface as Request;
require 'vendor/autoload.php';

$app = new \Slim\App;
$app->get('/webhooks/answer', function (Request $request, R

});

$app->post('/webhooks/events', function (Request $request,

});
```

```
$app->run();
```

This code creates a new `slim` application and defines two routes. These routes correspond to the following webhook endpoints:

- **`/webhooks/answer`**: Nexmo's APIs make a **GET** request to this endpoint when you receive an inbound call on your virtual number.
- **`/webhooks/events`**: Nexmo's APIs make a **POST** request to this endpoint every time a significant event occurs (such as call **ringing**, being **answered** and **completed**) to update your application on the status of the call.

## Log Call Events

Let's deal with the `/webhooks/events` endpoint first. Every time we are notified about an event we want to log it. This will help you debug any issues later on.

This example uses PHP's built-in web development server and the `error_log()` function to output data to a log file (`event.log`).

Create a `php.ini` file in the root of your application which includes the following settings:

```
error_log = ./event.log
```

```
log_errors = on
date.timezone = UTC
```

---

Then, update the `/webhooks/events` handler to log incoming event data:

```
$app->post('/webhooks/events', function (Request $request,
    error_log($request->getBody()));
});
```

## Handle Inbound Calls

When you receive an inbound call, Nexmo's API makes a GET request to your `/webhooks/answer` endpoint and expects the response to contain instructions on how to process the call.

You provide these instructions in the form of a Nexmo Call Control Object (NCCO) in JSON format. The NCCO defines the various "actions" that the call needs to take, such as `input` to collect any digits the user might press on their telephone keypad or `stream` to play audio into the call. You can find the full list of actions in the [NCCO Reference](#).

Before we concern ourselves with the proxy logic, let's first ensure that we can receive an inbound call on our Nexmo number.

Test your application with a talk action that uses TTS (text-to-speech) to read a message to your caller. Define and return the following NCCO in your `/webhooks/answer` route handler:

```
$app->get('/webhooks/answer', function (Request $request, Response $response) {  
    $ncco = [  
        [  
            'action' => 'talk',  
            'text' => 'Thank you for calling. Everything ap  
        ],  
    ];  
    return $response->withJson($ncco);  
});
```

## Make Your Webhooks Accessible

For Nexmo's APIs to make requests to your webhook endpoints they must to be accessible over the Internet.

A great tool for exposing your local development environment to the public Internet is ngrok. Our [tutorial](#) shows you how to install and use it.

Launch ngrok using the following command:

Make a note of the public URLs that ngrok created for you. These will be similar to (but different from) the following:

```
http://066d53c9.ngrok.io -> localhost:3000
```

```
https://066d53c9.ngrok.io -> localhost:3000
```

---

On their free plan, every time you restart ngrok the URLs change and you will have to update your Voice API application configuration. So leave it running for the duration of this tutorial.

## Purchase a Number

You need a Nexmo virtual number to receive phone calls. If you don't already have a Nexmo number you can purchase one easily using the [Vonage CLI](#).

You can install the Vonage CLI using the [Node Package Manager](#), npm:

```
npm install -g vonage-cli
```

---

Then, configure the Vonage CLI with your API key and secret from the [developer dashboard](#):

```
vonage config:set --apiKey=YOUR_API_KEY --apiSecret=YOUR_AP
```

To see what numbers are available, use `vonage number:search`, passing it your two-character country

code. For example, GB for Great Britain or US for the USA. You want to ensure that the number you purchase is able to receive voice calls:

```
vonage number:search COUNTRY_CODE --features=VOICE
```

---

Choose a number from the list and buy it using the following command:

```
vonage number:buy <NUMBER>
```

---

You will be prompted to confirm your purchase. Make a note of the number that you bought.

## Create a Nexmo Voice API Application

You now need to create a Nexmo Voice API Application. An application in this context is not the same as the application you have just written the code for. Instead, it is a container for the configuration and security information you need to use the Voice API.

You'll use the Vonage CLI again for this. You need to specify the following information:

- A name for your application
- The public URL to your `/webhooks/answer` endpoint (e.g. `https://066d53c9.ngrok.io/webhooks/answer`)

- The public URL to your `/webhooks/events` endpoint (e.g. `https://066d53c9.ngrok.io/webhooks/events`)
- The name and location of the file that will contain your security credentials

In the same directory as your PHP application, enter the following command, supplying the appropriate URLs for your webhook endpoints:

```
vonage apps:create "My Voice Proxy" --voice_answer_url=ANSW
```

This configures a Voice API application with your webhooks. The Voice Application is identified by a unique application ID: make a note of this as you will need it in the next step. You'll also want to copy your private key, which you can view by running:

## Link the Application to Your Nexmo Number

Next, you need to link your Voice API application to your Nexmo number.

Execute the following command, replacing `APPLICATION_ID` with the one generated by the `vonage apps:create` command that you executed in the preceding step:

```
vonage apps:link APPLICATION_ID --number=NEXMO_NUMBER
```

---

Verify that the number and application are linked by executing the `vonage apps:show` command. You can also see this information in the [developer dashboard](#).

## Call Your Nexmo Virtual Number

With `ngrok` running on port 3000 in one terminal window, launch your PHP application on the same port in another:

```
php -S localhost:3000 -c php.ini
```

---

Call your Nexmo number. If everything is working OK, you should hear the message you defined in your NCCO and then the call terminates.

Check the `event.log` and note which events were recorded during the call.

## Implement the Proxy Logic

To create a voice proxy we can connect the inbound call to our target recipient's number and use our Nexmo virtual number to hide the inbound caller's real number.

Create two string constants to store these numbers:

- **FROM\_NUMBER**: Your Nexmo virtual number
- **TO\_NUMBER**: A landline or mobile number you can use

for testing. This should be different from the number you will use to make the initial call.

```
define(FROM_NUMBER, );  
define(TO_NUMBER, ;
```

---

Both numbers should include the country code and omit any leading zeroes. For example, in the UK the country code is 44. So if my mobile number is 07700 900004, the correct format is 447700900004.

Use the NCCO connect action to connect the inbound caller to the recipient. Replace the current NCCO in your `/webhooks/answer` endpoint with the following:

```
$ncco = [  
  [  
    'action' => 'connect',  
    'from' => FROM_NUMBER,  
    'endpoint' => [  
      [  
        'type' => 'phone',  
        'number' => TO_NUMBER,  
      ],  
    ],  
  ],  
];
```

---

## Test the Voice Proxy

Ensure that ngrok is still running in one terminal window and re-launch your PHP application in another:

```
php -S localhost:3000 -c php.ini
```

---

Call your Nexmo virtual number. When the call is answered it should immediately ring your second number. That call should originate from your Nexmo number rather than the number you used to place the call.

That's it! You have now implemented a simple voice proxy that hides a caller's real number.

## Further Reading

If you want to learn more about the Voice API, check out the following resources:

- [Voice API overview](#)
- [Voice API reference](#)
- [NCCO Reference](#)