

How to Host a Laravel App with MySQL Database on Heroku



Orie Chinedu Emmanuel

Oct 20, 2017 · 5 min read

This article outlines a step-by-step approach to hosting a Laravel App on Heroku. It also explains how to make your app on Heroku work with MySQL database since ordinarily, Heroku does not make provision for MySQL. I saw the need to write about this topic after undergoing series of challenges to host my Laravel test app with MySQL database on Heroku.

Requirements

In this article, I assume that you already have your Laravel App ready for deployment. The following tools will be required:

- i) Command prompt for windows or its equivalent for other Operating System users.*
- iii) A Heroku account. If you don't have one click [here](#) to create an account*
- iv) Heroku command line interface (CLI), Heroku has its CLI for interacting with it. You can download one [here](#).*

Intro

Heroku is one of the platforms as a service that offers developers free hosting services. This enables a developer test the performance of his/her application before moving it to the final production environment. One of the drawbacks with this platform is that it does not offer MySQL Server by default but of course, there's a way around it as there are other platforms that also offer free Mysql servers such as [db4free](#) (which will be used in this article) or [ClearDB addons](#) which complements this shortfall. Follow along as I explain how you can host your Laravel app on Heroku and using the MySQL Server service from other platforms.

Step 1

Open the command prompt. Navigate to the directory where your Laravel project is located. (Note git bash does not support Heroku CLI). Login to your Heroku account by running;

```
heroku login
```

Step 2

Run the following commands:

```
git init //to initialize a new repo
git add --all
git commit -m "my first commit"
```

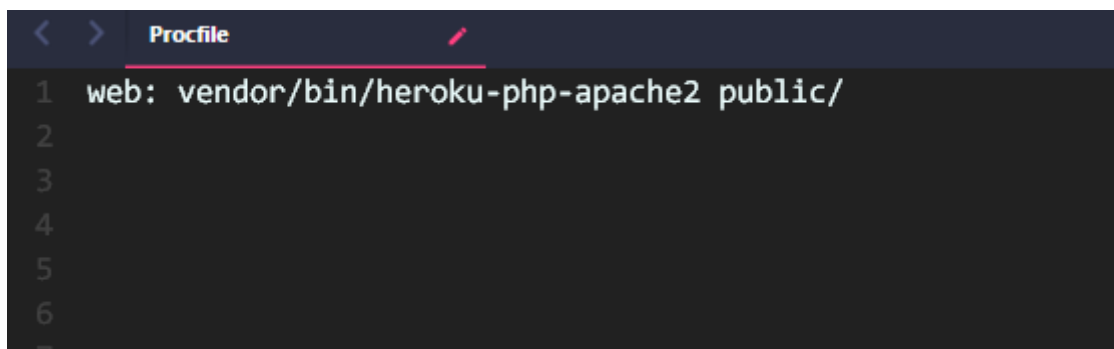
Step 3

Create a *Procfile*. Heroku launches an Apache server with PHP to serve the app from the root of the project which in our case is the `public/` folder. A *Procfile* is required to configure it.

Create a file at the root of your project and name it *Procfile*.

Note no extension is required and the letter P must be uppercase. Copy the string below and paste in the Procfile you created and save it.

```
web: vendor/bin/heroku-php-apache2 public/
```



The Procfile Content should look like this

Alternatively, you can run the command below to create the *Procfile*

```
echo "web: vendor/bin/heroku-php-apache2 public/" > Procfile
```

Then run,

```
git add Procfile  
git commit -m "Heroku Procfile"
```

Next is to create the new Heroku app.

Step 4

To create your heroku app run;

```
heroku create <myappname>
```

Note `<myappname>` above serves as your domain name. If you don't add it, Heroku will generate an app name for you.

Next is to specify the language of the app as PHP.

Step 5

To create a buildpack that will specify PHP as the app language run;

```
heroku buildpacks:set heroku/php
```

Next is to generate an app key for the Laravel app on Heroku environment.

Step 6

Run `php artisan key:generate --show`

Copy the generated key and run the command below setting the APP_KEY value to the key you copied.

```
heroku config:set APP_KEY = <key generated above>
```

We need to set the error log to Heroku error log. To do that, go to config/app.php and set `'log'=>'errorlog'`. With this, you can view your app error log on your Heroku dashboard. Save and git add and commit.

It's time to host the app!

Step 6

To host the app to heroku, run

```
git push heroku master --app <myappname>
```

Note: the part is `--app <myappname>` only necessary if you have more than one app hosted on Heroku.

To launch the app from command line run

```
heroku open --app <myappname>
```

If your only goal is to host your Laravel app without a database, then you can ignore next section and jump to the conclusion.

. . .

Integrating Mysql Database to the App

Our next target is to add a Mysql database to our app. There are several sites that offer free Mysql server. But for this article, we'll be using [db4freee.net](https://db4free.net). The steps are as follows:

Step 1

Sign up on db4free.net. After signing up, you'll be prompted by mail to verify your email. After verification, you'll be redirected to a page where you'll see the following data:

```
Database: [database name you specified]

Username: [username you specified]

Email:    [your email]
```

Step 2

Go to your Laravel project and navigate to `config/database.php` and update it with the db4free MySQL database details you created earlier as shown below:

```
'mysql' => [
    'driver' => 'mysql',
    'host' => env('DB_HOST', 'db4free.net'),
    'port' => env('DB_PORT', '3306'),
    'database' => env('DB_DATABASE', 'your_database_name
from_db4free'),
    'username' => env('DB_USERNAME', 'your_db4free_
username'),
    'password' => env('DB_PASSWORD',
'your_db4free_password'),
    'unix_socket' => env('DB_SOCKET', ''),
    'charset' => 'utf8mb4',
    'collation' => 'utf8mb4_unicode_ci',
    'prefix' => '',
    'strict' => true,
    'engine' => null,
],
```

Save it. git add , commit and push to Heroku.

```
git add config/database.php
git commit -m "update database"
git push heroku master --app <myappname>
```

Step 4

You need to use the Heroku bash in order to run your migrations. Run,

```
heroku run bash
```

Next is to run your migrations, ensure you're now running subsequent commands on the the Heroku bash.

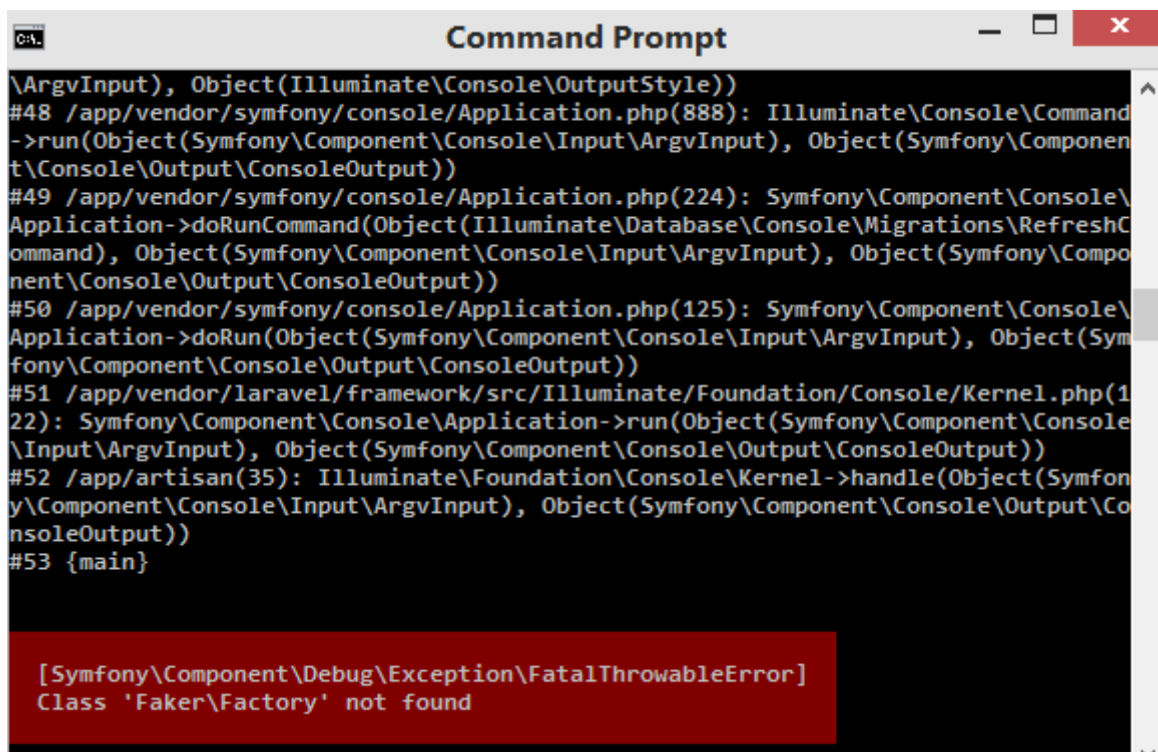
```
php artisan migrate --app <myappname>
```

At this point, your database is set. If you want to seed the database then you can proceed to the next steps as follows:

Step 5

```
php artisan db:seed --app <myappname>
```

If you encounter an error similar to the screenshot below, do not panic, you've got covered.



To fix this, follow the steps below:

- Go to your laravel project, navigate to composer.json
- Replace "fzaninotto/faker" of "require-dev" with "require"
- Exit the heroku bash by running `exit`
- Git add and commit the changes to your project

```
git add composer.json
git commit -m "modified composer.json"
git push heroku master --app <myappname>
```

- Go back to heroku bash,
- Run `heroku run bash`
- Run `composer install`
- After the composer has finished installing the necessary dependencies, proceed to migrate and seed your database by running the commands below

```
php artisan migrate:refresh --seed --force
```

Boom! everything now works fine!

Summary

You have now learned how to host your awesome Laravel app on Heroku. You have also learned how you can incorporate MySQL server to enable your app work with the MySQL database. If you'd want to learn more about Heroku, you can visit <https://devcenter.heroku.com/>. If you also want to learn about auto-deploy to Heroku from Git click [here](#).

Thanks for taking your time to read, I do hope the article has helped you!

Medium

About Help Legal