

Deploying your application to an external web host (Heroku)

This guide walks through steps I followed to deploy the CRUD examples to Heroku hosting and also used external databases (making the site completely external to any local development environment).

A couple quick notes: this assumes you've been following all guides/tutorials up until this point configuring the Node.js and PHP projects.

Part 1. Node.js

I'm following the guide posted here: <https://devcenter.heroku.com/articles/deploying-nodejs>

Note that this guide assumes you already have your development environment set up with npm and Node.js.

1. Create a free Heroku account
 - a. <https://signup.heroku.com/>
2. Install the Heroku CLI
 - a. <https://devcenter.heroku.com/articles/heroku-cli>
 - b. Windows users may need additional steps
 - i. If you get path error, manually add the path to your environment variables
 1. <https://stackoverflow.com/questions/44301515/heroku-cli-installation-error-path-not-updated-original-length-1585-1024/47726413#:~:text=You%20should%20also%20try%20to,will%20manually%20update%20the%20PATH.>
 - ii. If you want to run Heroku via the powershell terminal run
 1. `npm i -g heroku`
 2. `Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy Unrestricted`
 - iii. Verify with `heroku -v`
3. I assume you have a properly configured Node.js project using npm init.
4. Modify your port to include "`process.env.PORT ||`" so it will work with Heroku's dynamic port settings e.g.,

```
const PORT = process.env.PORT || 8080;
```

<https://help.heroku.com/P1AVPANS/why-is-my-node-js-app-crashing-with-an-r10-error>

5. Test that you can run the application:

```
$ heroku local web
```

Your app should now be running on <http://localhost:5000/>.

6. Following the above guide, make sure you are using a `.gitignore` file to prevent unnecessary files being pushed to the final repository (e.g., you don't want your `node_modules` or `.vscode` included in commits).

7. Per the above guide, log in to Heroku using your credentials
 - a. `heroku login`
 - b. `heroku create`
 - i. Note: if you type this exactly, a URL will be generated for you. If you type something after create, this will be a part of your URL. For example, typing `heroku create cpsc314-example` would create a url with that as the subdomain: `https://cpsc314-example.herokuapp.com`
 - c. `git push heroku master`
 - i. Note: This assumes that you have a local git repository on a master branch and that you have made commits to your local repository.
 - d. `heroku open`
 - i. this opens up the app in your browser
8. **NOTE: you may need to configure your database settings to match an accessible database OR have a fallback so the site loads without database access before this application will run.**
9. After changes, you need to commit your changes then re-run 7c and 7d

Part 2. MongoDB

We will make use of a free online MongoDB host for this step.

1. Create a free MongoDB account or use a MongoDB of your choice.
 - a. <https://www.mongodb.com/pricing>
 - b. Create a cluster and a database and any collections just as you would with a local db.
 - c. If you select a cluster, then click connect it will walk you through creating your connection string.
2. If you want to limit IP access:
 - a. <https://www.mongodb.com/blog/post/integrating-mongodb-atlas-with-heroku-private-spaces>

Part 3. PHP

The same process is followed for PHP but a few extra steps before pushing to heroku.

<https://devcenter.heroku.com/articles/getting-started-with-php?singlepage=true#deploy-the-app>

1. Make sure to install heroku
 - a. `sudo snap install heroku --classic`
2. You also need to install composer
 - a. `sudo apt install composer`
3. You may also need a PHP buildpack

<https://elements.heroku.com/buildpacks/heroku/heroku-buildpack-php>

You'll need to use at least an empty `composer.json` in your application.

```
$ echo '{}' > composer.json
$ git add composer.json
$ git commit -m "add composer.json for PHP app detection"
```

4. Now you should be able to follow step 7c in the Nodejs example.

Part 4. MySQL

We will make use of a free online MySQL host for this step. I make no claims to be familiar with these! Just something to note.

Possible choices:

1. This is the one I tested with
 - a. <https://www.freemysqlhosting.net/> (this one worked immediately though it seems the 'free' account expires and you have to sign up again with different credentials later)
 - b. <https://www.db4free.net/>
 - i. This one seemed to work but was pretty slow.
2. Did not try
 - a. <https://remotemysql.com/#about> (took a bit to get a reply for creating this account – sketchy, survey required to create a database)
3. Once you have access to a MySQL database, you will want to run the table creation scripts to get started.
 - a. You can do this in phpMyAdmin for DB 1
 - b. Alternately, you can remotely connect using MySQL Workbench and do the same.