

Heroku Checklist:

First, create your new Git repo and app:

1. Create a new Git Repo with your app name, e.g. testing-heroku. Click 'Initialize with a README'.
2. Clone the repo to your computer.
3. `cd` into testing-heroku
4. Create a new Express app in the testing-heroku folder by running the command `express . --view=hbs --git`
5. Say `y` to destination is not empty, continue? [y/N].
6. Make sure to add a `.gitignore` file to your project and ignore `node_modules/` and `.env`
7. Run `npm install`
8. Add and commit your changes.
9. Push the changes up to your origin master branch.
10. Check `git remote -v -->` You should only see origin available.
11. Run `heroku create` to add a new heroku origin
12. Run `git remote -v` again to check that you now have both origin and heroku upstreams available.
13. Run `git push heroku master` to push your latest commit to Heroku.
14. Check your deployed link by copying the link from your logs into your browser OR running `heroku open`
15. You can also check <https://dashboard.heroku.com/apps/> to see your running apps

Then, hook up your Mongo database:

YOU MAY HAVE TO VERIFY YOUR PAYMENT INFO WITH HEROKU FIRST! YOU WILL NOT BE CHARGED.

11. Run `heroku addons:create mongolab:sandbox` to create a new Mongo DB.
12. Check `heroku config` to view your environment variables.
13. You should see an environment variable called `MONGODB_URI`.
14. `npm install --save mongoose`
15. `npm install --save dotenv`
16. If you would like to reference this URI in your code, you can use `process.env.MONGODB_URI`.
17. Create a `.env` file in the root of your application on your local machine and add `MONGODB_URI=your local mongodb server url` to it.
(e.g. `MONGODB_URI=mongodb://localhost/my_database_name`)

18. Add `require('dotenv').config();` to the top of your `app.js`.

19. Add these lines to your `app.js` or `server.js` file:

```
var mongoose = require('mongoose');  
mongoose.connect(process.env.MONGODB_URI);
```

20. Make sure you've added your `.env` file to your `.gitignore` file.

21. COMMIT AND SAVE

Then, set up Nodemon

19. `npm install --save nodemon`

20. In the `scripts` object in your `package.json`, add `"dev": "nodemon ./bin/www"` below the `start` script.

21. Now, when we want to run Nodemon as our dev server, we just have to type `npm run dev`.