**You should not hardcode port in Express application.**

That is why you will see the following in the “www” script of express generated project. The port will read from the environment.

var port = normalizePort(process.env.PORT || '3000');

**So use :**

app.listen(process.env.PORT);

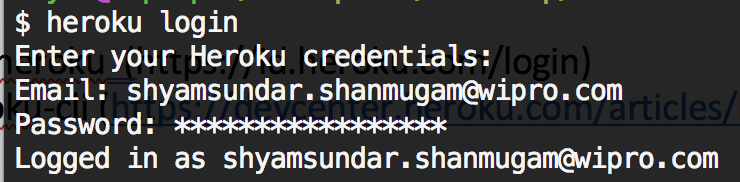
**For accessing from Angular Service, you should not use**[**http://localhost:5000**](http://localhost:5000/)**. When you are doing development, you can test by running Angular using “ng serve” which will have its own developmental web server and Express will be another web server.   But when you want to deploy in production, you should not do that way. You should do a production build  of Angular and then copy it to Express public directory like explained in ILP-003 tips document.**

**Because Angular application is running within Express server in production, you need not use absolute URL  [http://localhost:5000/employees/create](http://localhost:5000/employees/create" \t "_blank) ). Instead use relative URL ( i.e., “/employees/create”)**

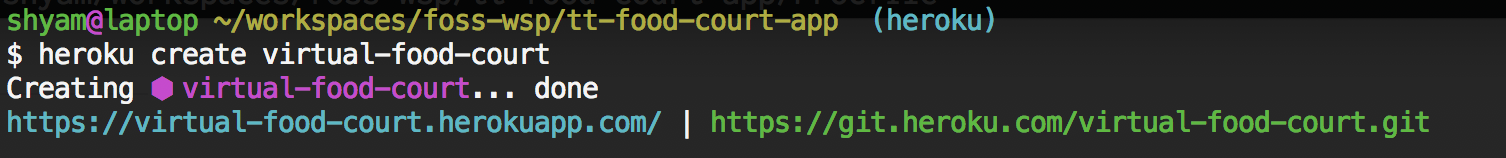
**In case of MongoDB, you cannot use your local MongoDB server. Use can use a MongoDB server on cloud like mLab (or Mongo Atlas – Mongo Atlas has acquired mLab and sometime in future, mLab may get merged into Mongo Atlas ). For now, you can use mLab which provides a free tier. When you create an account and create a database, the URL details to access will be provided which you can take and use in your Express application.**

# Steps to deploy to Heroku

1. Sign up to heroku (<https://id.heroku.com/login>)
2. Install heroku-cli (<https://devcenter.heroku.com/articles/heroku-cli>)
3. Open CMD terminal and type “**heroku**” to check installation.
4. Login 🡪 “**heroku login”**



1. Create new app 🡪 “**heroku create APPNAME**”



1. **NOTE** : cd to Node.js app dir where “**package.json**” exists and “**public**” dir contains PRODUCTION build files of ANGULAR/REACT
2. **In app’s root directory , create a file that is named Procfile without a file extension. This file tells Heroku which command(s) to run to start your app. These commands are probably the same as the ones you use to run your code on your local machine.**

**an example Procfile for a simple Node.js app:**

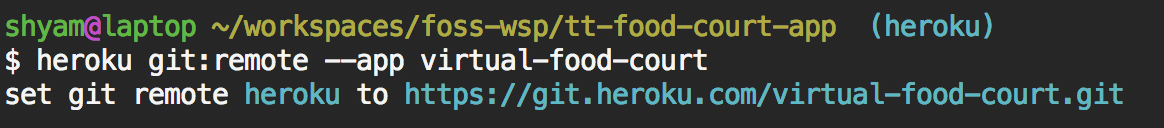
web: node app.js

**NOTE : Be sure to initialize the Git repository in your Node.js app’s root directory. If your app is in a subdirectory of your repository, it won’t run when it is pushed to Heroku.**

1. Create .git dir 🡪 **git init**

* **git add .**
* **git commit –m “COMMIT MESSAGE”**

1. Create git remote 🡪 “**heroku git:remote --app APPNAME”**



1. Push to heroku 🡪 “**git push heroku master**”



1. Set the config variable(s) if any , example : to connect to mongo



1. **Visit the app URL.**