RANGLE.IO

Implementing a full-stack solution using Express and create-react-app

by Rob Brander

About me

```
const presenter = {
  name: 'Rob Brander',
  email: 'rob.brander@rangle.io',
  github: 'rbrander'
}
```

Background

I wanted to build a React app and host it on Heroku

- `create-react-app` was designed for building a front-end solution
- Using Node with Express, I could provide a back-end solution
- Heroku can be used to host your application for free
- Challenge: To deploy a full-stack react app using only one heroku process

Using `create-react-app`

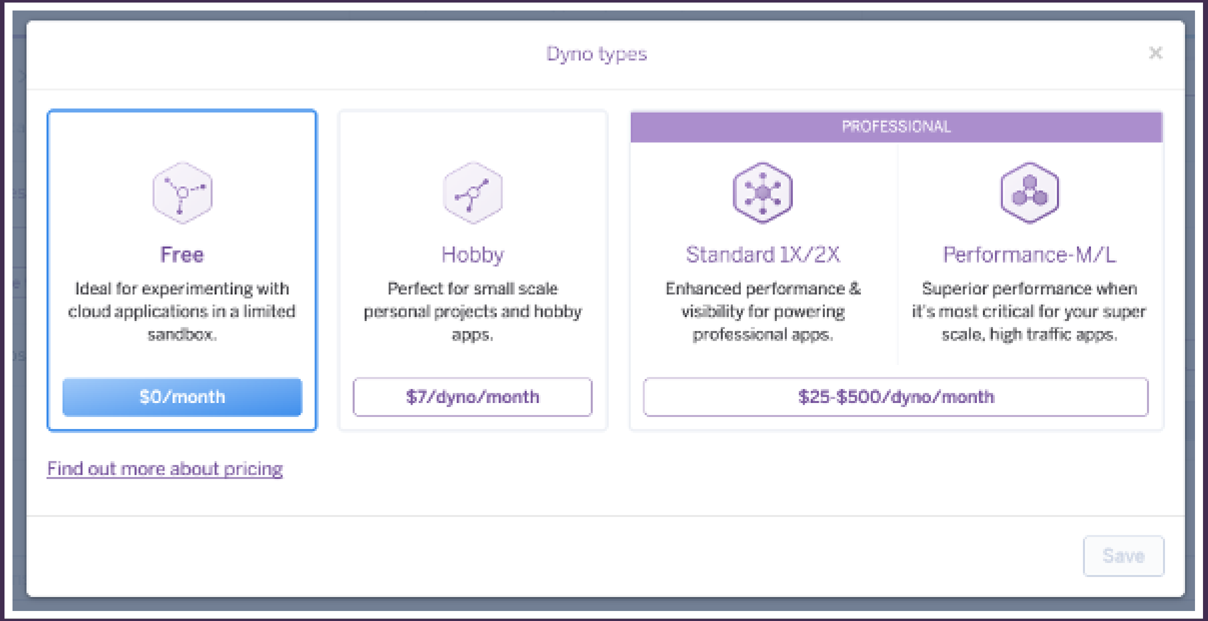
- Installation: `npm install create-react-app -g`
- Create a project: `create-react-app myProject`
- Run dev server: `npm run start`
- Build for production: `npm run build`

Setting up Heroku

- Push source code to github
- Create a heroku instance
- Bind the github repo to the heroku instance
- Deploy the master branch
- Dyno runs `npm start`
- Heroku now running dev server

Heroku Dynos

- Dynos are processes that are run on your Heroku instance
- Web and Worker processes
- Free account permits one web process OR one worker process
- All HTTP traffic is sent to only the 'web' process



Dyno types

PROFESSIONAL

Free	Hobby	Standard 1X/2X	Performance-M/L
Ideal for experimenting with cloud applications in a limited sandbox.	Perfect for small scale personal projects and hobby apps.	Enhanced performance & visibility for powering professional apps.	Superior performance when it's most critical for your super scale, high traffic apps.
$0/month	$7/dyno/month	$25-$500/dyno/month	

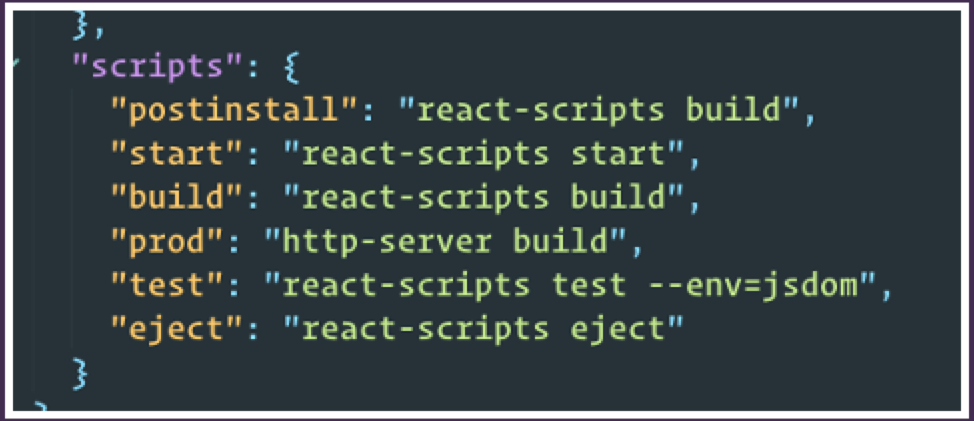
Find out more about pricing

Production deploy

Instead of deploying the dev server, I wanted to deploy the production build and run it using a web server.

- Need to run `npm run build` on each deploy
- Run a web server to host the files in `build/`
- Setup Procfile to run `npm run prod`

```
$ echo "web: npm run prod" > Procfile
```



```
  },
  "scripts": {
    "postinstall": "react-scripts build",
    "start": "react-scripts start",
    "build": "react-scripts build",
    "prod": "http-server build",
    "test": "react-scripts test --env=jsdom",
    "eject": "react-scripts eject"
  }
```

Backend

Simple Express Server

Options:

- Run another site all together using a heroku instsance and a separate repo (CORS)
- Run the server as a 'worker' process (pay for the extra dyno)
- or find some way to merge the back-end with the front-end

The Solution

- The production code was already built
- Let the back-end host the front-end, statically!

```
app.use(express.static('./build'));
```

- Add prefix for API routes
- Update the `npm run prod` to run the back-end server

Open challenge to the audience

Find a way to run the back-end server and the front-end using npm start (for dev)

Thank you

```
const presenter = {
  name: 'Rob Brander',
  email: 'rob.brander@rangle.io',
  github: 'rbrander'
}
```