

Deploy your React app on Heroku

井民全, Jing, mqjing@gmail.com

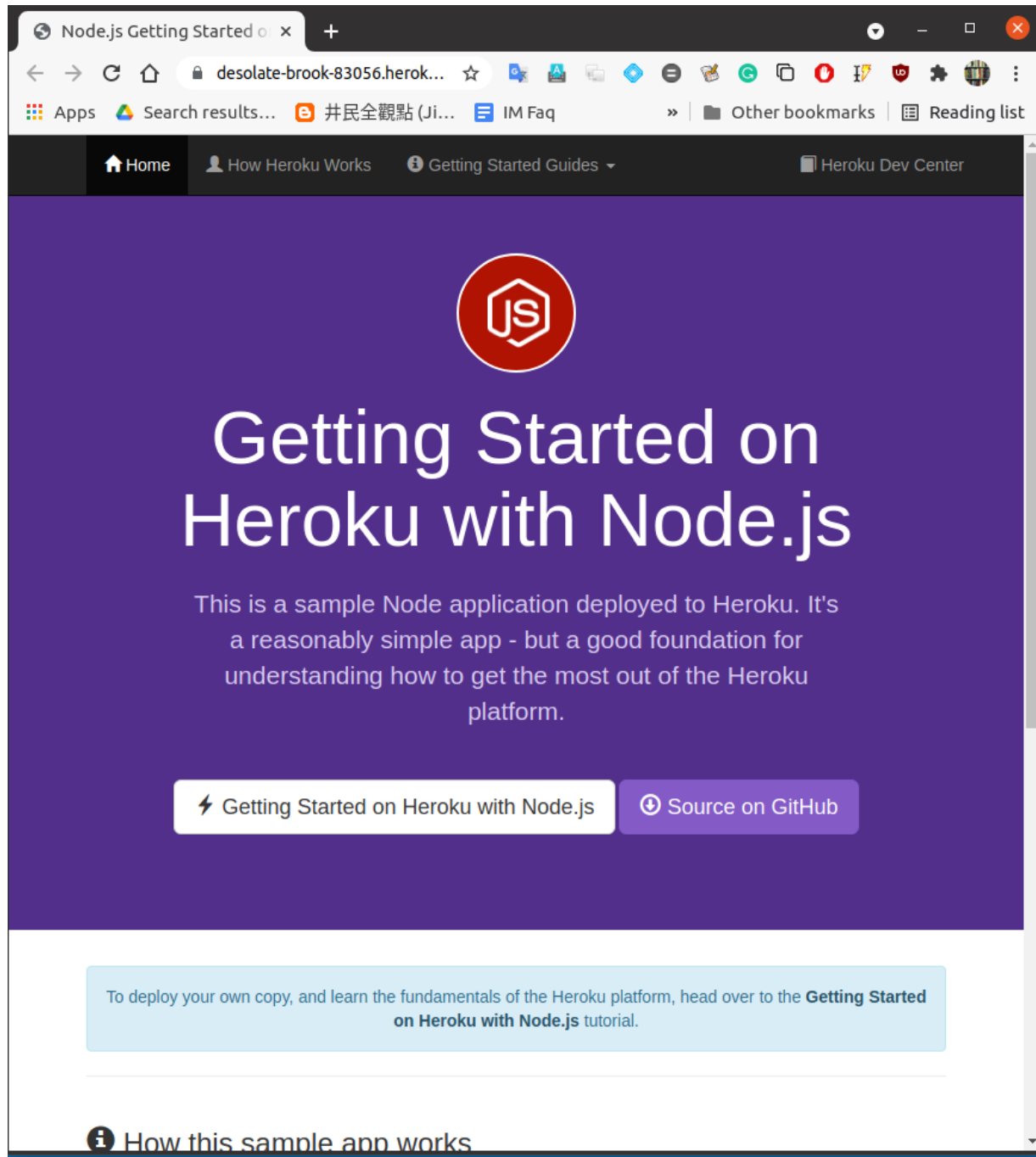
Here, I'll show you how to deploy a HelloWorld nodejs app on [Heroku](#). A React app deployment also includes forming a useful guide. For a free dyno, the container running your app sleeps after 30 mins of inactivity^[1]. However, it's enough to give a trial to test your idea.

Enjoy!

By Jing.

Table of contents

1. Setup the environment	6
2. Sample Project	6
2.1. Prepare the sample project	6
2.2. Deploy the project	6
2.3. Visit your app	9
2.4. Log	10
3. Hello word	11
3.1. Modify Code	11
3.2. Commit to the local repository	11
3.3. Deploy to heroku	11
3.4. Visit your app	12
4. React app	12
4.1. Create the client	12
4.2. Link to the express	12
4.3. Deploy to heroku	13
4.4. Visit your React app	13
4.5. Delpy Log	14
5. References	14
6. FAQ	14



Quick

Template

```
# Setup CLI
## Ubuntu
sudo snap install heroku --classic

## Mac
brew install heroku/brew/heroku

heroku login

# Prepare your project
git clone https://github.com/heroku/node-js-getting-started.git
cd node-js-getting-started

# Deploy
heroku create           # Create a heroku project associate to your project
git push heroku main   # Deploy the project
heroku open             # Visit your app

# Log
heroku logs --tail
```

Deploy React app

```
mkdir client
cd client
npx create-react-app my-app --template typescript # create a React sample application

yarn build # bundle the react client
```

Link to the Express

```
const express = require('express')
const path = require('path')
const PORT = process.env.PORT || 5000

const app = express();
// mount the static middleware to the client path
app.use(express.static(path.join(__dirname, 'client/my-app/build')));

// route the index.html to the client index.html
app.get('/', function (req, res) {
```

```
res.sendFile(path.join(__dirname, 'client/my-app/build/index.html'));  
});
```

```
app.listen(PORT, () => console.log(`Listening on ${ PORT }`));
```

Deploy

```
git add -f client/my-app/build # The React bundle client
```

```
git add client/my-app
```

```
git commit -m 'Add React client'
```

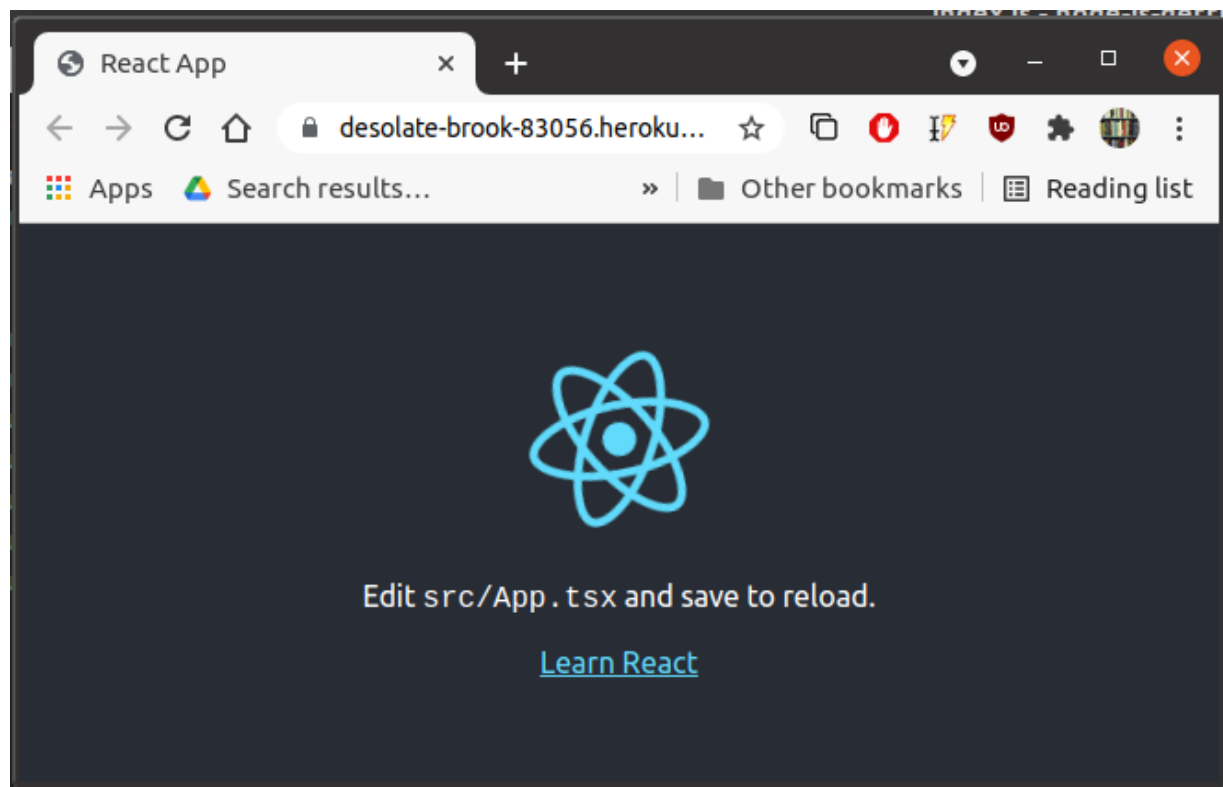
```
git push heroku main # Deploy
```

Visit the React app

```
heroku open
```

Log

```
heroku logs --tail
```



Detail

1. Setup the environment

Step 1: Install the CLI tool

```
sudo snap install heroku --classic
```

Step 2: Log in to the Heroku CLI

```
heroku login
```

2. Sample Project

2.1. Prepare the sample project

Clone the sample project

```
git clone https://github.com/heroku/node-js-getting-started.git
```

2.2. Deploy the project

To deploy your project to the heroku server, you can use the command **heroku create** that will generates a random name for your app.

Step 1: Create a heroku project associate to your project

```
cd node-js-getting-started
heroku create
```

```
jing@jing-VirtualBox:~/test/heroku/01-helloworld/node-js-getting-started$ heroku
create
> Warning: heroku update available from 7.56.0 to 7.56.1.
Creating app... done, desolate-brook-83056
https://desolate-brook-83056.herokuapp.com/ | https://git.heroku.com/desolate-br
ook-83056.git
```

Fig. Create a heroku project associates to your project.

Verify

<https://dashboard.heroku.com/apps>

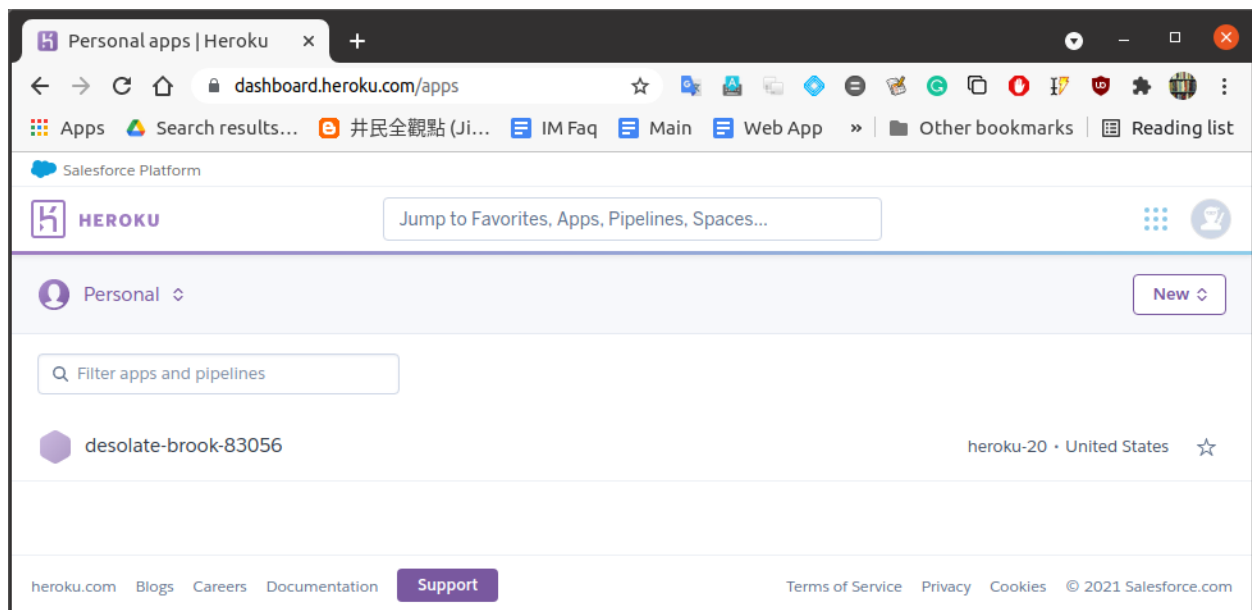


Fig. Check the created project.

Check out the branch status.

Command

git branch -a

```
jing@jing-VirtualBox:~/test/heroku/01-helloworld/node-js-getting-started$ git branch -a
* main
  remotes/heroku/main
  remotes/origin/HEAD -> origin/main
  remotes/origin/main
jing@jing-VirtualBox:~/test/heroku/01-helloworld/node-js-getting-started$
```

Fig. The branch status.

Check out the git config status

Command

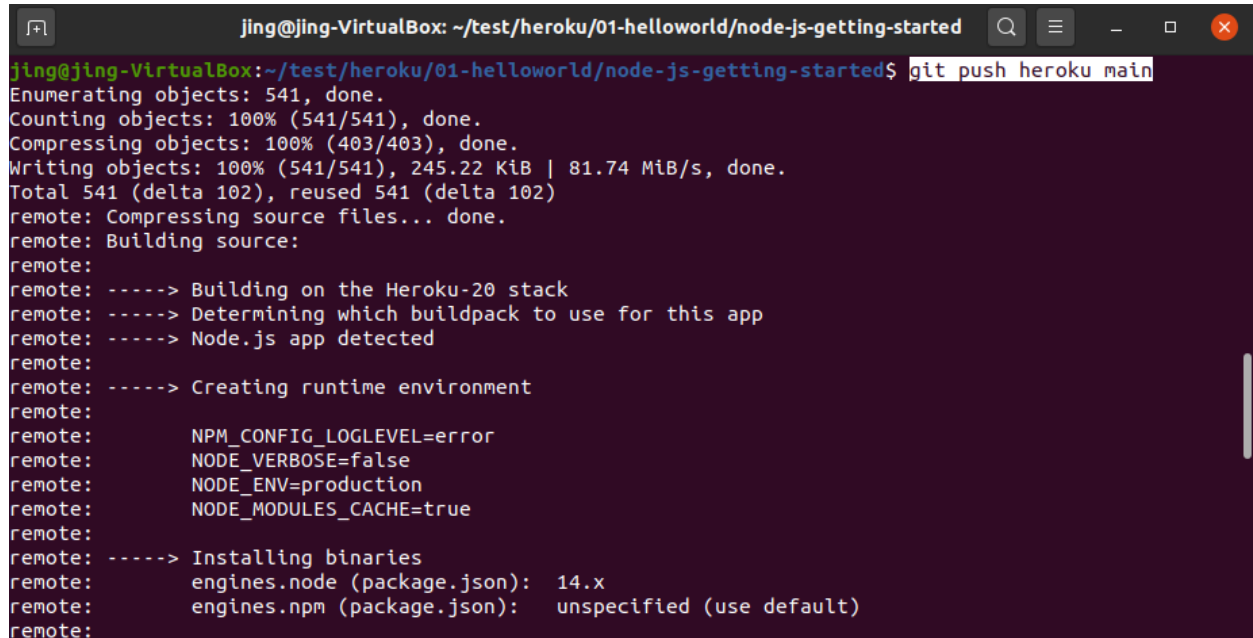
cat .git/config

```
[core]
  repositoryformatversion = 0
  filemode = true
  bare = false
  logallrefupdates = true
[remote "origin"]
  url = https://github.com/heroku/node-js-getting-started.git
  fetch = +refs/heads/*:refs/remotes/origin/*
[branch "main"]
  remote = origin
  merge = refs/heads/main
[remote "heroku"]
  url = https://git.heroku.com/desolate-brook-83056.git
  fetch = +refs/heads/*:refs/remotes/heroku/*
```

Step 2: Deploy to heroku

Just push your code to the remote repository **heroku** with main **branch**. The heroku server will do the rest tasks.

```
git push heroku main # push the code to the main branch in repository remotes/heroku
```

A terminal window titled 'jing@jing-VirtualBox: ~/test/heroku/01-helloworld/node-js-getting-started' shows the execution of the command 'git push heroku main'. The output displays the progress of pushing the code to Heroku, including object enumeration, counting, compressing, and writing. It then shows the remote server's response, indicating the building of the source on the Heroku-20 stack, determining the buildpack (Node.js), and creating the runtime environment. Finally, it shows the installation of binaries for Node.js (14.x) and npm (unspecified).

```
jing@jing-VirtualBox: ~/test/heroku/01-helloworld/node-js-getting-started$ git push heroku main
Enumerating objects: 541, done.
Counting objects: 100% (541/541), done.
Compressing objects: 100% (403/403), done.
Writing objects: 100% (541/541), 245.22 KiB | 81.74 MiB/s, done.
Total 541 (delta 102), reused 541 (delta 102)
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Building on the Heroku-20 stack
remote: -----> Determining which buildpack to use for this app
remote: -----> Node.js app detected
remote:
remote: -----> Creating runtime environment
remote:
remote:       NPM_CONFIG_LOGLEVEL=error
remote:       NODE_VERBOSE=false
remote:       NODE_ENV=production
remote:       NODE_MODULES_CACHE=true
remote:
remote: -----> Installing binaries
remote:       engines.node (package.json):  14.x
remote:       engines.npm (package.json):   unspecified (use default)
remote:
```

Fig. Deploy your project by git push.

2.3. Visit your app

```
heroku open
```

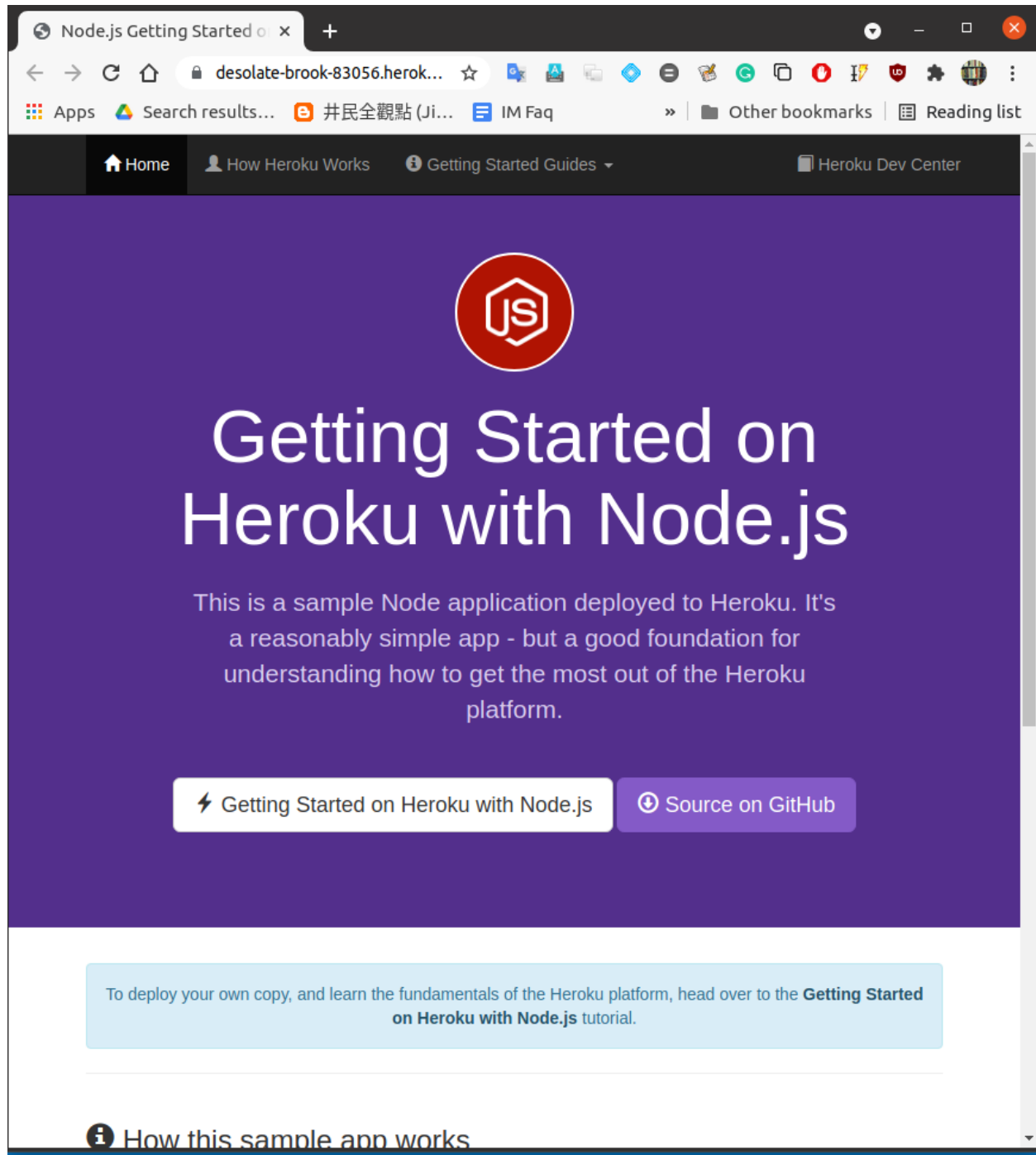


Fig. The sample project.

2.4. Log

```
heroku logs --tail
```

3. Hello word

3.1. Modify Code

File: views/pages/index.ejs

```
<!DOCTYPE html>
<html>
<head>
  <%- include ("../partials/header.ejs") %>
</head>

<body>

  <%- include ("../partials/nav.ejs") %>

  Hello World.

</body>
</html>
```

3.2. Commit to the local repository

```
git add views/pages/index.ejs
git commit -m "Hello world"
```

3.3. Deploy to heroku

```
git push heroku main
```

3.4. Visit your app

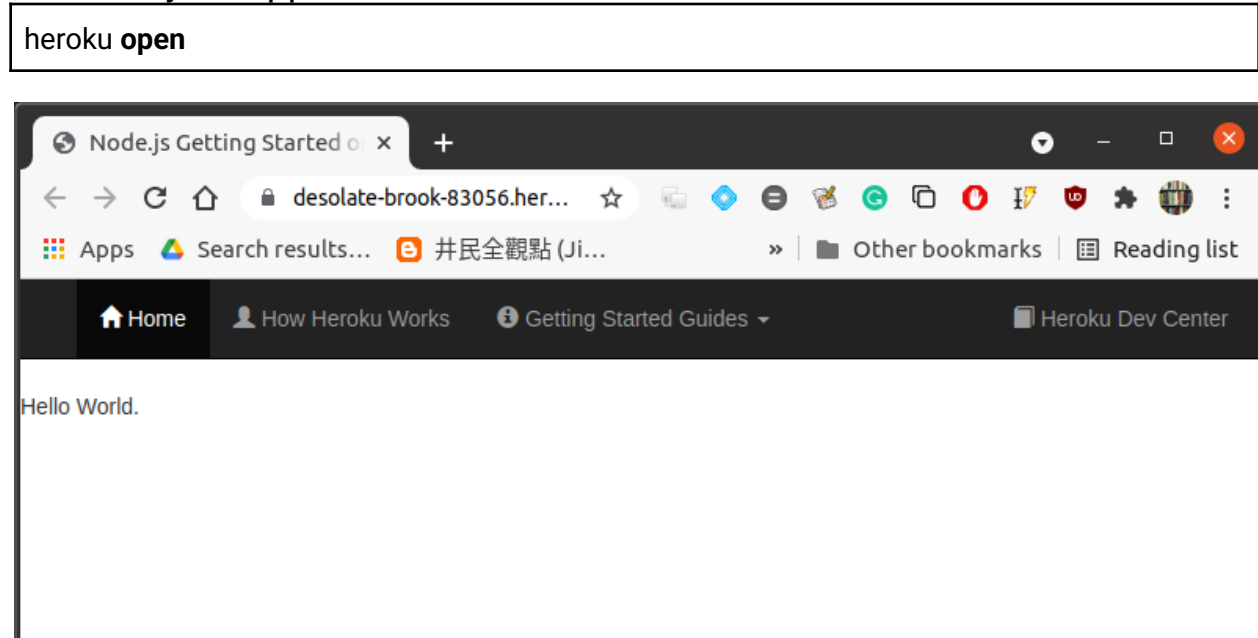


Fig. Hello World on heroku server.

4. React app

4.1. Create the client

```
# create a React sample application
mkdir client
cd client
npx create-react-app my-app --template typescript

# test the react
cd my-app
npm start

# bundle the react client
yarn build # it will generate a folder named ./build
```

4.2. Link to the express

File: index.js

```
const express = require('express')
const path = require('path')
const PORT = process.env.PORT || 5000

const app = express();
// mount the static middleware to the client path
app.use(express.static(path.join(__dirname, 'client/my-app/build')));

// route the index.html to the client index.html
app.get('/', function (req, res) {
  res.sendFile(path.join(__dirname, 'client/my-app/build/index.html'));
});

app.listen(PORT, () => console.log(`Listening on ${ PORT }`));
```

4.3. Deploy to heroku

```
# The React bundle client
git add -f client/my-app/build

# [Option] whole React source (nevermind the node_modules, it will be skipped)
git add client/my-app

git commit -m 'Add React client'

# Deploy
git push heroku main
```

4.4. Visit your React app

```
heroku open
```

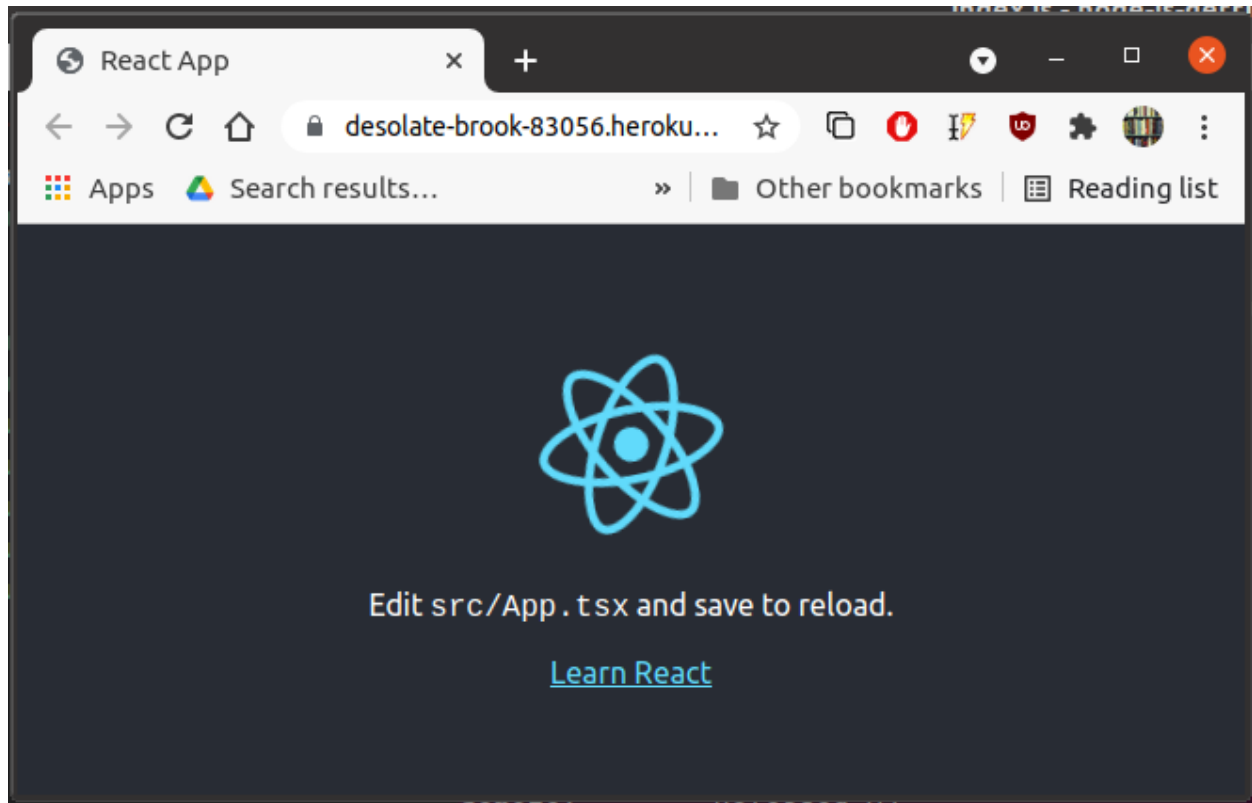


Fig. The React app.

4.5. Delpy Log

```
heroku logs -tail
```

5. References

1. <https://www.heroku.com/pricing>
2. <https://devcenter.heroku.com/articles/getting-started-with-nodejs#set-up>

6. FAQ

1. [\[cloud, main\] The Cloud FAQ](#)

2. [\[frontend, main\] The React, Angular, WebAPI, Chart FAQ](#)
3. [\[backend\] The NodeJS, Express, Koa, Web FAQ](#)