

---

# 01UDFOV/01TXYOV – WEB APPLICATIONS I

## DEPLOYMENT ON THE CLOUD

### WHAT ARE WE BUILDING THIS WEEK?

In this lab, we will deploy the web-based task manager developed during BigLab2 on the cloud. Specifically, you will launch the app on a *Platform as a Service* cloud platform called **Heroku**. In this manner, the web-based task manager you have implemented will be accessible for everyone on the Web. The following are the instructions you should follow to register on Heroku, configure the deployment environment, and launch the application.

### SETTING UP THE HEROKU CLI

- Create a Heroku account (each member of the group). To do so, you must go to their site and create a free account (<https://signup.heroku.com/login>).
- Download and install the Heroku Command Line Interface (CLI); it enables you to create and manage your Heroku apps directly from the terminal. The simplest way to install the Heroku CLI is by running the command `npm install -g heroku`<sup>1</sup>.
- Once you installed the CLI, run in the terminal the `heroku login` command. You will have to press *any* key (and then press “enter”) to go to your web browser to complete the login. The CLI will then log you in automatically.

### SETTING UP YOUR PROJECT

- Create a new branch in your Big Lab 2 repository (e.g., `deploy`) and pull it.
- Re-arrange the structure of your BigLab2 project. Move out all the files within the `server` folder to the `root` of your project, then delete the empty `server` folder.
- In the `package.json` file that you have just moved in the `root`, add the following line inside the scripts `"heroku-postbuild": "cd client && npm install && npm run build"`.

Your file should look like:

```
"scripts": {
  "start": "node server.js",
  "heroku-postbuild": "cd client && npm install && npm run build",
  "test": "echo \"Error: no test specified\" && exit 1"
},
```

- Inside the client `package.json` remove the line related to the proxy.

---

<sup>1</sup> However, feel free to try other installation methods indicated in the official documentation (<https://devcenter.heroku.com/articles/heroku-cli>)

- To access properly the client folder from the server, it is necessary to install the [path module](#)<sup>2</sup>. To do so, open another terminal and install the **path** module (**npm install path**), then import it in server.js (i.e., `const path = require('path')`).
- In the same file, modify the **port** constant by setting:
 

```
const PORT = process.env.PORT || 3001;
```
- Additionally, indicate Express how to serve static files by adding:
 

```
app.use(express.static("./client/build"));
```
- Finally, **below all the endpoint definitions**, add a new endpoint to declare that any request that does not match any other endpoints send back the client React application index.html file. The modifications in your **server.js** file should look like the snippet below.

```
const path = require('path');
const port = process.env.PORT || 3001;

app.use(express.json());
app.use(express.static("./client/build"));

[...]

app.get('*', (req, res) => {
  res.sendFile(path.join(__dirname, 'build', 'index.html'));
});
```

## DEPLOYING THE APPLICATION ON HEROKU

- Open the terminal, step up in the root of your project, and execute the following commands:

```
heroku create
git add --all
git commit -m "Preparing deploy to Heroku"
git push heroku <your_new_branch>:<heroku_main_branch>
```

- To determine whether the `<heroku_main_branch>` is master or main, run the command **git branch --all**. It shows both the local and the remote branches of your repository. The starred branch (\*) is your current branch. You should see also a remote branch related to Heroku called "main" or "master".
- After executing them, Heroku will install all the required node modules and will create and deploy an instance with your web-based task manager. Run the command **heroku open** to open the application in the browser.

### Hints:

1. You can find the deployment of a complete working sample application at: <https://github.com/mars/heroku-cra-node>
2. You can check your logs for details from the Heroku CLI with the command: **heroku logs --tail**

---

<sup>2</sup> The path module provides utilities for working with file and directory paths.