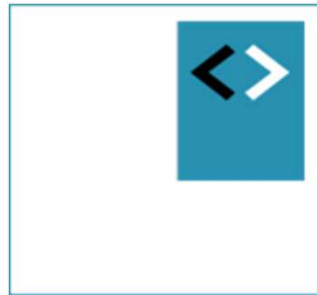
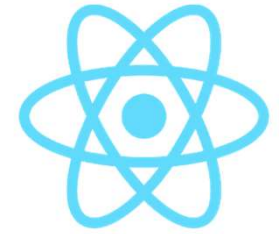


React Fundamentals

Module - Deployment

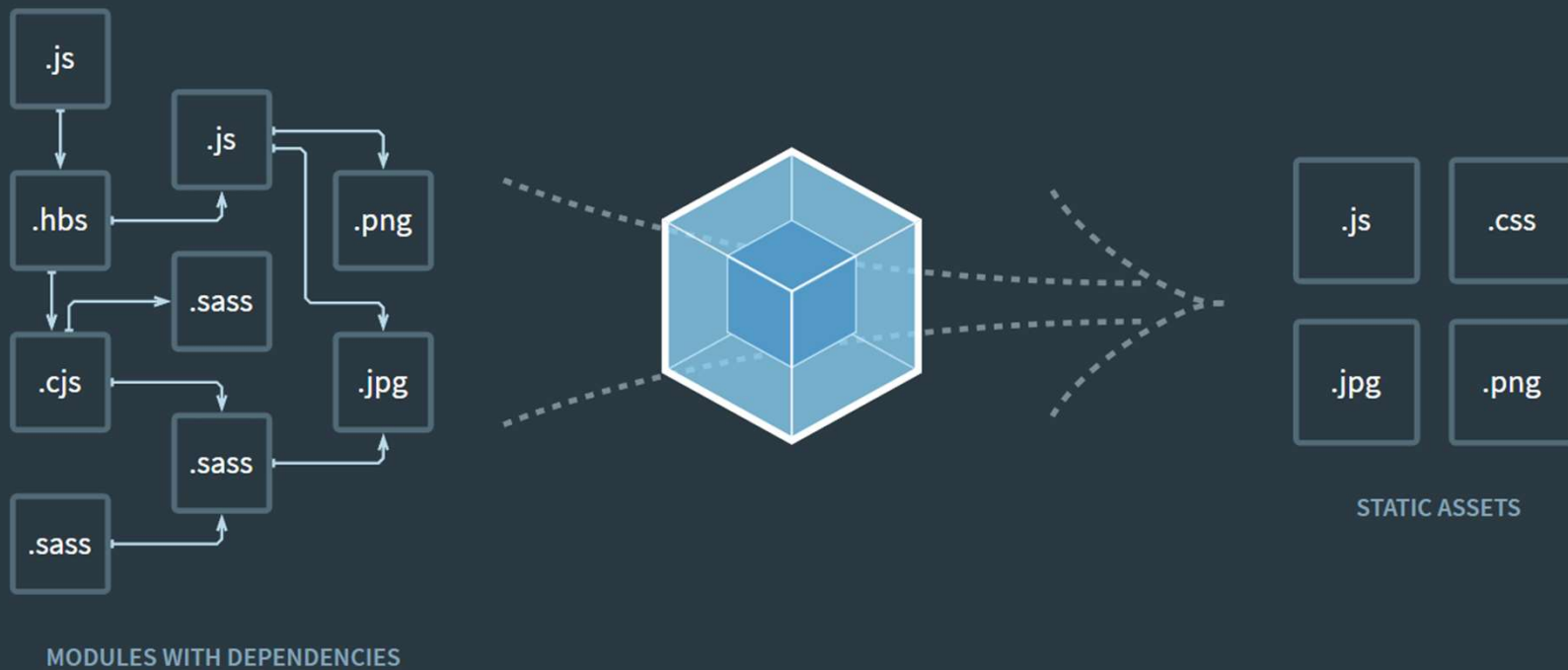


Peter Kassenaar –
info@kassenaar.com



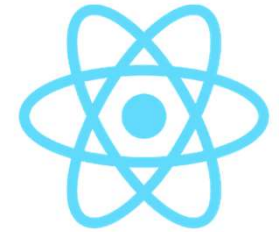
"npm run build creates a build directory with a production build of your app. Inside the build/static directory will be your JavaScript and CSS files. Each filename inside of build/static will contain a unique hash of the file contents."

bundle your scripts



<https://webpack.js.org/>

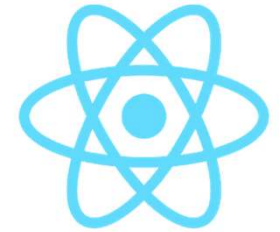
Deployment



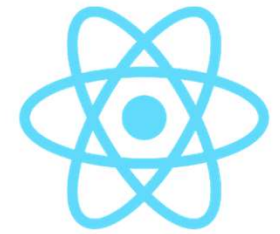
- So, deployment is a task of **Create React App**
- CRA has a script prepared for that

```
npm run build
```

Local http-server



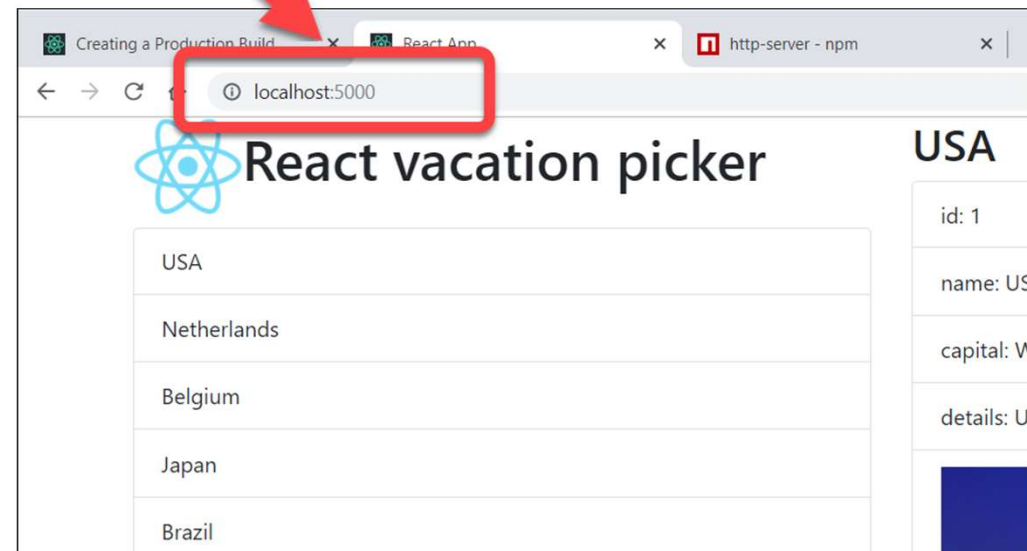
- Deploy the contents of the `/build` directory to a webserver
- Local, command line webserver; options:
- `npm install serve -g`
 - `cd` into the directory and type `serve -s`
 - Content is hosted at `http://localhost:5000`
- `npm install http-server -g`
 - `cd` into the `build` directory and type `http-server`
 - Content is hosted at `http://localhost:8080`



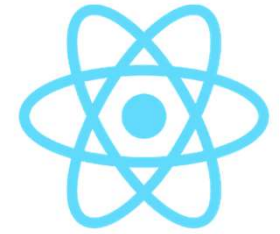
```
C:\Windows\System32\cmd.exe - serve
Microsoft Windows [Version 10.0.18362.476]
(c) 2019 Microsoft Corporation. Alle rechten voorbehouden.

C:\Users\Gebruiker\Desktop\react-fundamentals\training-app>cd /d %~dp0\build
C:\Users\Gebruiker\Desktop\react-fundamentals\training-app\build>serve

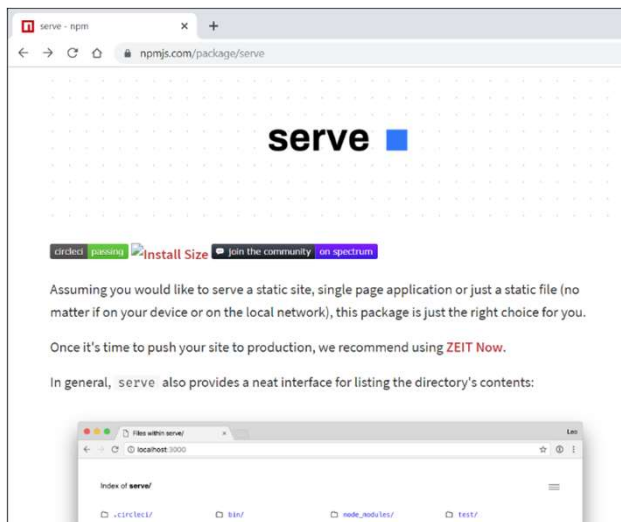
Serving!
- Local:      http://localhost:5000
- On Your Network: http://192.168.2.53:5000
Copied local address to clipboard!
```



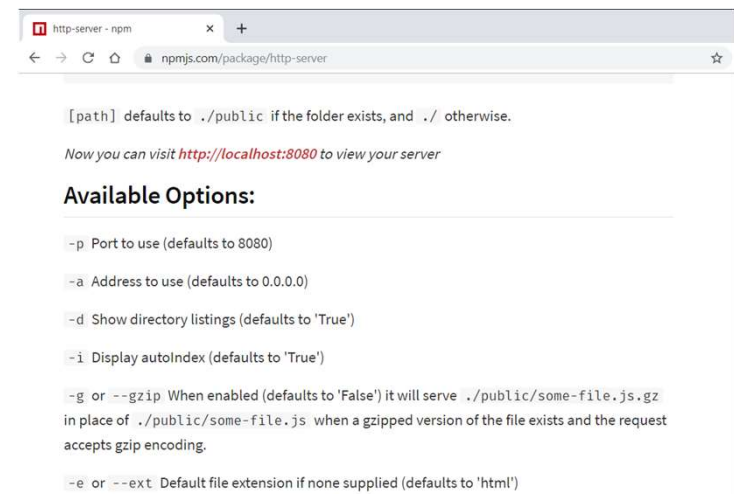
Options



- The flag `-s` redirects all requests to `index.html` (for a Single Page Application)
- See other options in online documentation

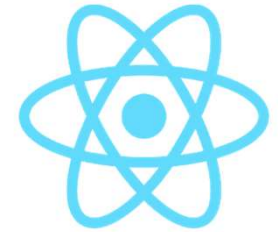


<https://www.npmjs.com/package/serve>

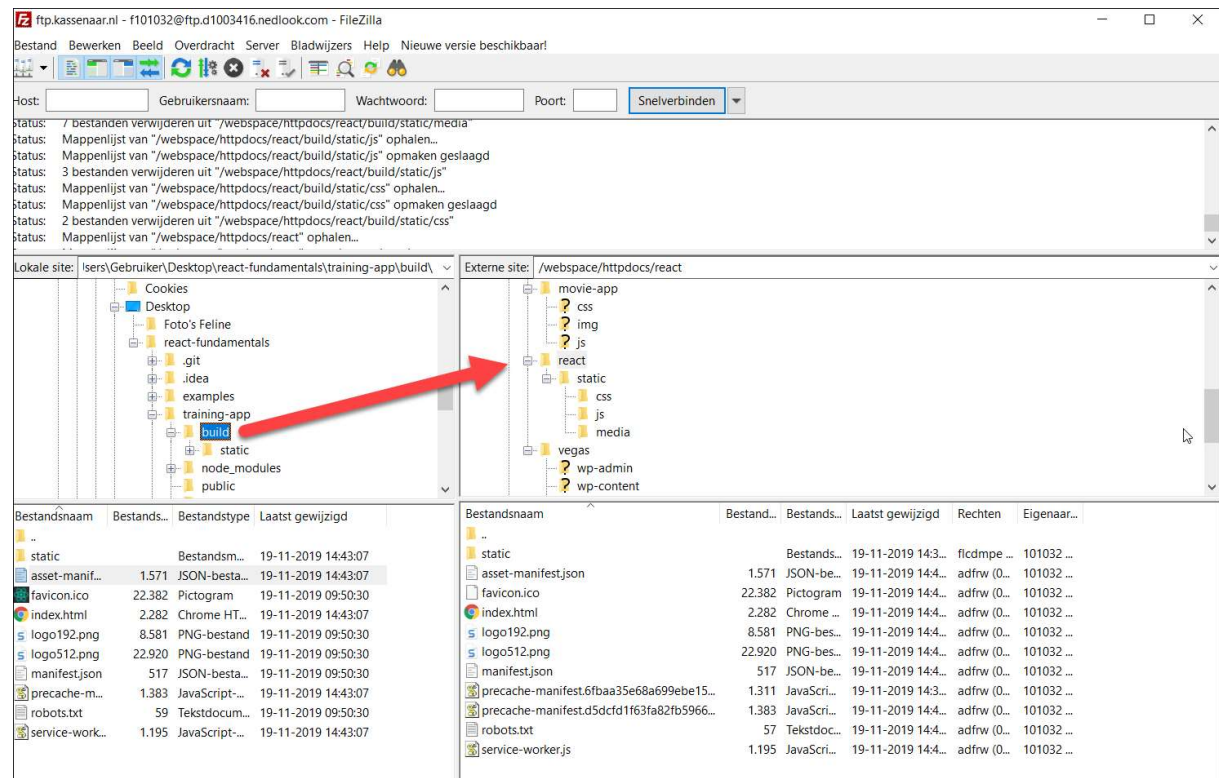
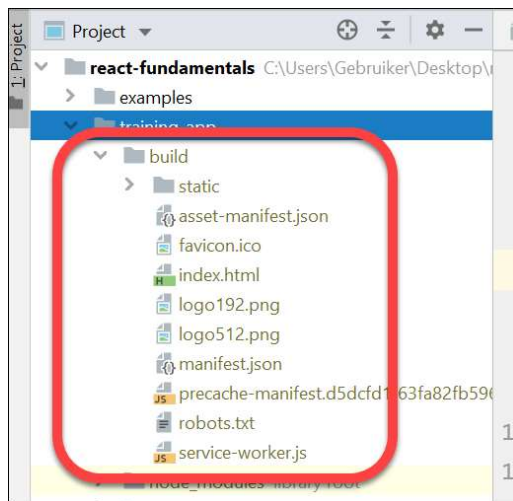


<https://www.npmjs.com/package/http-server>

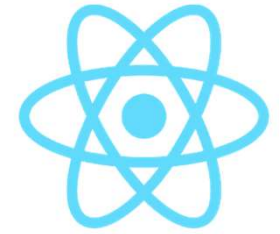
To online production server



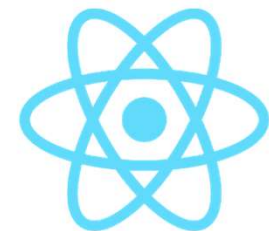
Just copy or FTP the contents of the `\build` folder to your production server



Deploying to a subfolder



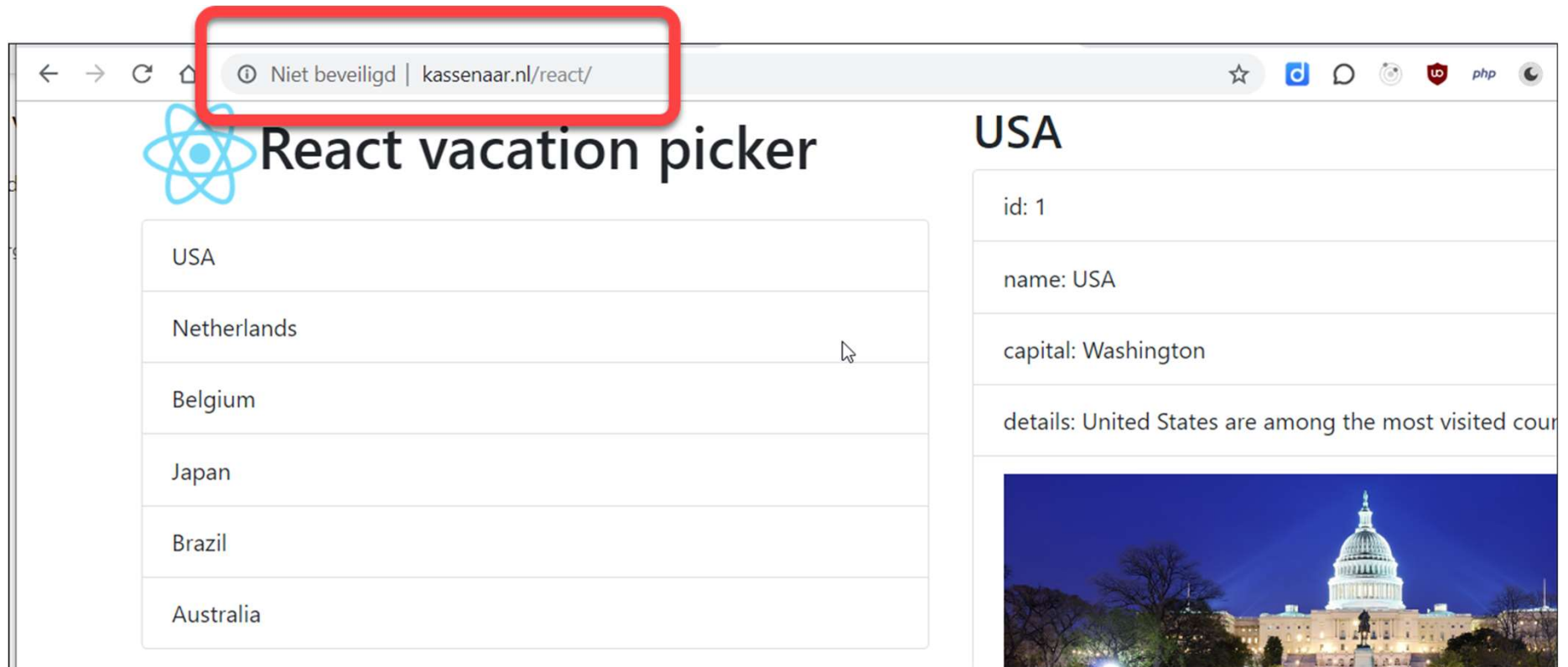
- Update the `<base href="/subFolderName/">` in your `index.html`
- Best practice:
 - 1. add a `"homepage": "name"` to `package.json`
 - 2. add `<base href="%PUBLIC_URL%/">` to `index.html`
- Build again
- Deploy to subfolder



```
package.json x index.html x
1 {
2   "name": "hello-react",
3   "version": "0.1.0",
4   "private": true,
5   "homepage": "/react/",
6   "dependencies": {
7     "bootstrap": "^4.3.1",
8     "jquery": "^3.4.1",
9     "popper.js": "^1.14.7",
10    "react": "^16.8.6",
11    "react-dom": "^16.8.6"
```

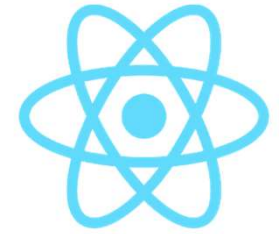
```
package.json x index.html x
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8" />
5     <base href="%PUBLIC_URL%/">
6     <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
7     <meta name="viewport" content="width=device-width, initial-scale=1" />
8     <meta name="theme-color" content="#000000" />
```

Result



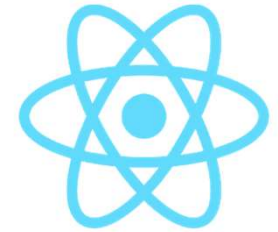
More info: <https://skryvets.com/blog/2018/09/20/an-elegant-solution-of-deploying-react-app-into-a-subdirectory/>


When using HTML5 mode for routing




- When using `history.pushState` (or HTML5 mode) for routing, the server needs to rewrite all requests to the `index.html` file
- This purely depends on the server you are using
 - Apache – use `mod_rewrite`
 - Nginx – use configuration rule
 - Internet Information Server – install `UrlRewrite` and update `web.config`
- See <https://create-react-app.dev/docs/deployment/#static-server> for detailed information
 - Lots of options and online hosting parties are described

Documentation on servers



 Create React App

Docs Help GitHub 

Welcome

Getting Started

Development

Styles and Assets

Building your App

Testing

Back-End Integration

Deployment

Deployment

Advanced Usage

Support

AWS Amplify

The AWS Amplify Console provides continuous deployment and hosting for modern web apps (single page apps and static site generators) with serverless backends. The Amplify Console offers globally available CDNs, easy custom domain setup, feature branch deployments, and password protection.

1. Login to the Amplify Console [here](#).
2. Connect your Create React App repo and pick a branch. If you're looking for a Create React App+Amplify starter, try the [create-react-app-auth-amplify starter](#) that demonstrates setting up auth in 10 minutes with Create React App.
3. The Amplify Console automatically detects the build settings. Choose Next.
4. Choose *Save and deploy*.

If the build succeeds, the app is deployed and hosted on a global CDN with an `amplifyapp.com` domain. You can now continuously deploy changes to your frontend or backend. Continuous deployment allows developers to deploy updates to their frontend and backend on every code commit to their Git repository.

Azure

See [this](#) blog post on how to deploy your React app to Microsoft Azure.

See [this](#) blog post or [this](#) repo for a way to use automatic deployment to Azure App Service.

Firebase

Static Server

Other Solutions

Serving Apps with Client-Side Routing

Building for Relative Paths

Serving the Same Build from Different Paths

Customizing Environment Variables for Arbitrary Build Environments

AWS Amplify

Azure

Firebase

GitHub Pages

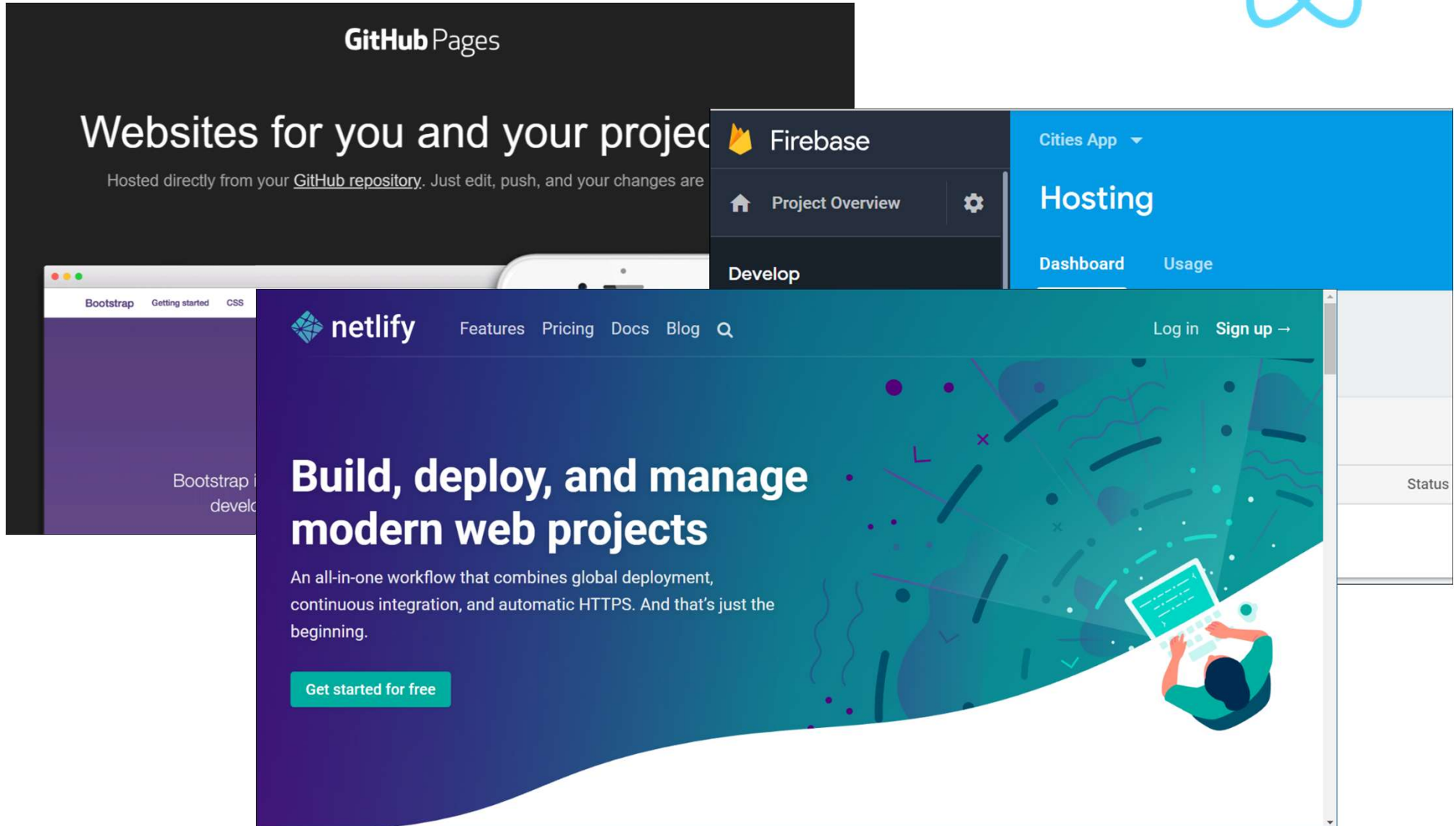
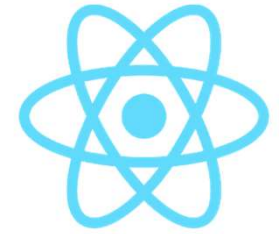
Step 1: Add homepage to package.json

Step 2: Install gh-pages and add deploy to scripts in package.json

Step 3: Deploy the site by running `npm run deploy`

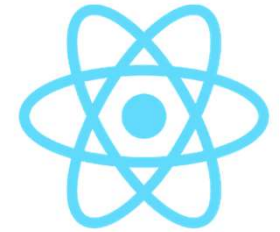
Step 4: For a project page, ensure your project's settings use gh-pages

Lots of free SPA-hosting sites available



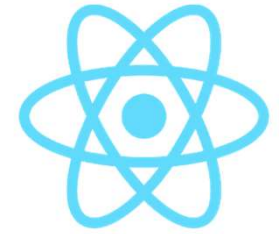
And many, many more...

Option: Netlify



A screenshot of the Netlify website homepage. At the top, a dark banner reads "New! Deliver web projects 10x faster. Get the free enterprise paper →". Below this is a navigation bar with the Netlify logo, links for "Products", "Pricing", "Docs", "Customers", and "Blog", a search icon, and links for "Contact sales", "Log in", and "Sign up →". The main content area features a large headline: "One workflow. From local development to global deployment." followed by a sub-headline: "Netlify is everything you need to build fast, modern websites: continuous deployment, serverless functions, and so much more." A green button labeled "Get started for free" is positioned below the sub-headline. The background of the main section is a dark teal with abstract geometric shapes and a stylized illustration of a person working on a laptop.

<https://www.netlify.com/>



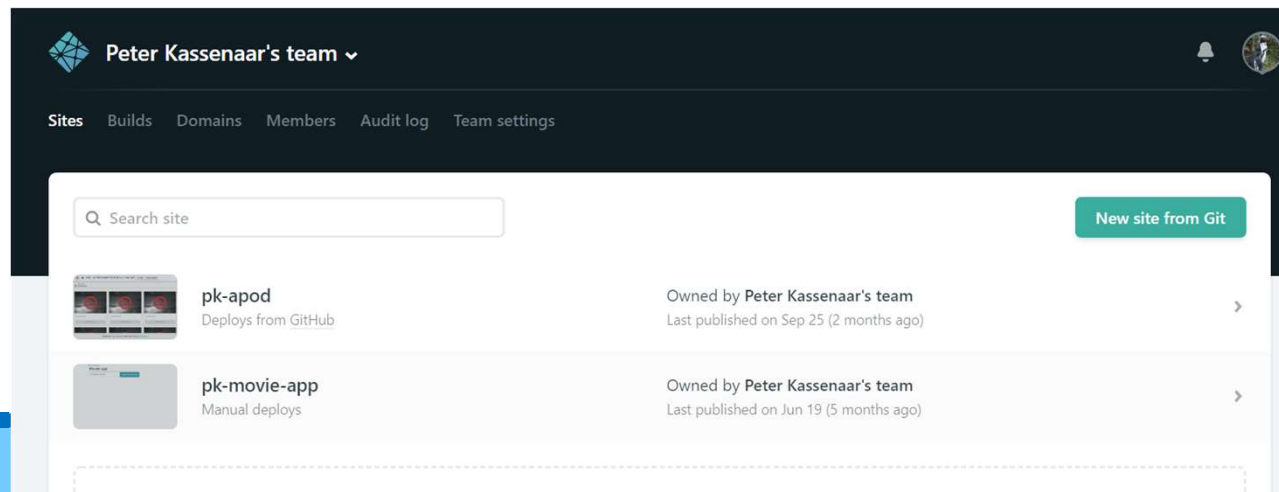
Two options

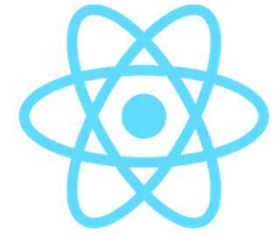
1. Manual build and deploy

- Drag and drop the `/build` directory

2. Continuous Integration

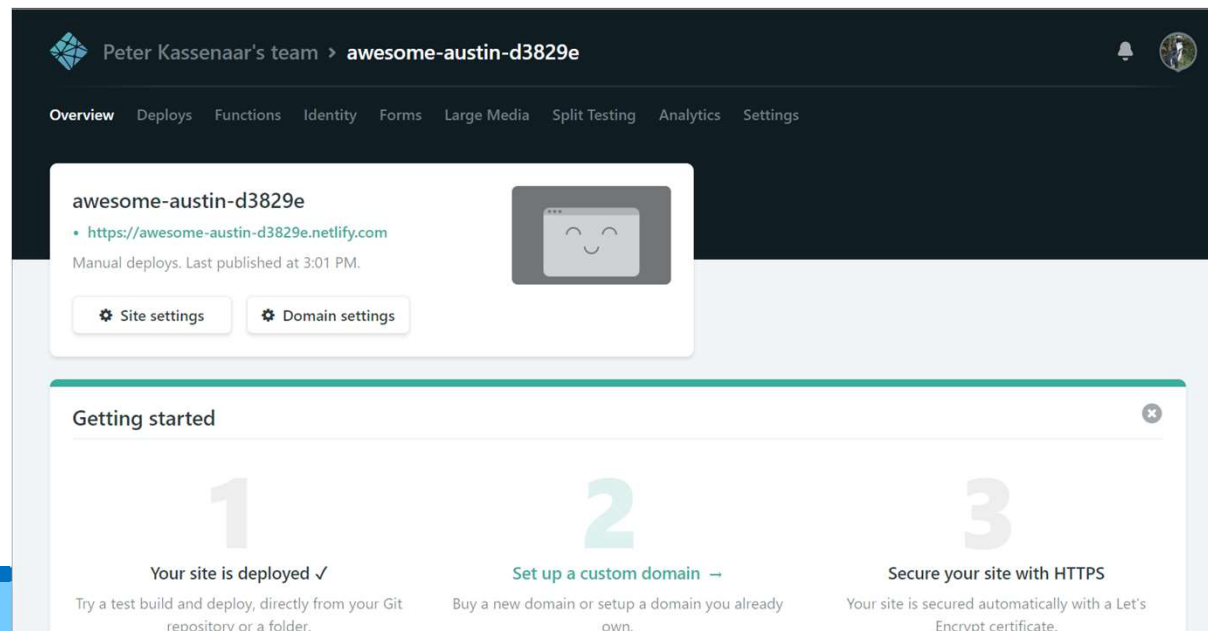
- Set up a Github Repo and connect Netlify to the repo
- If you push to the repo (or a specific branch), the application is rebuild and deployed



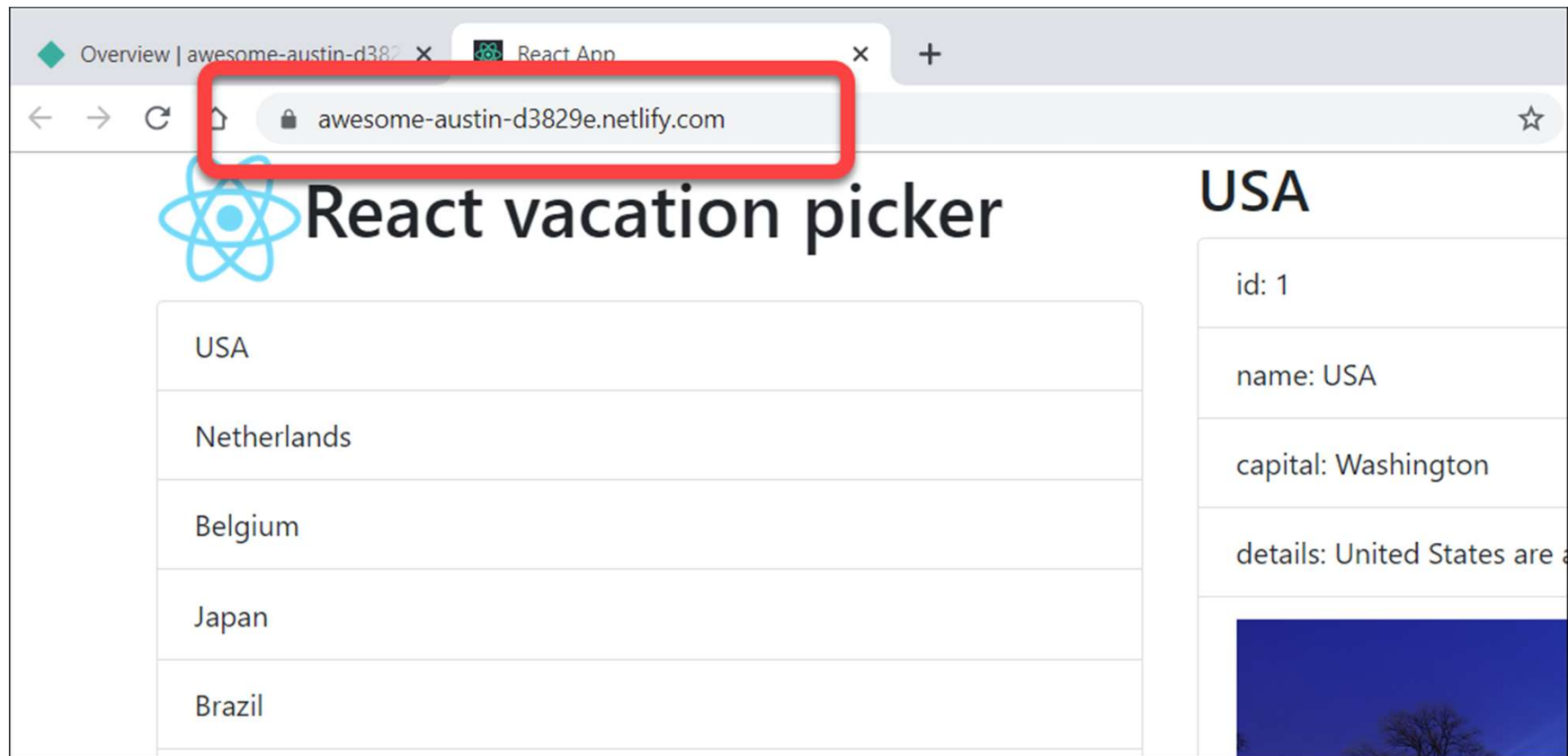


Easiest – Manual Deploy

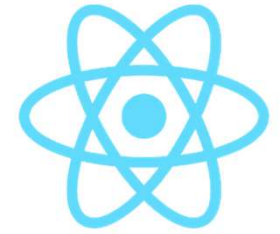
- Create a free Netlify account
- Just drag & drop the `/build` folder to Netlify dashboard
- Your project gets a random name



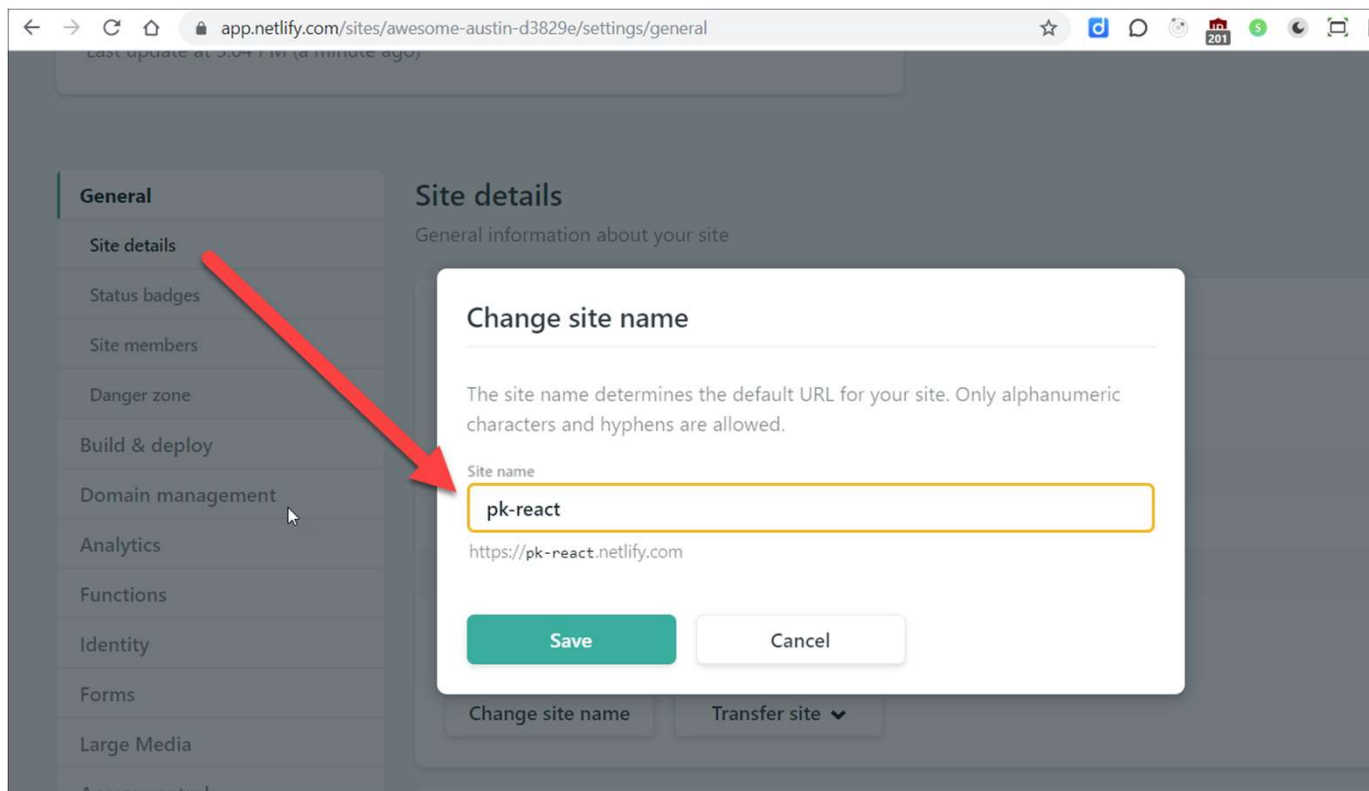
Result



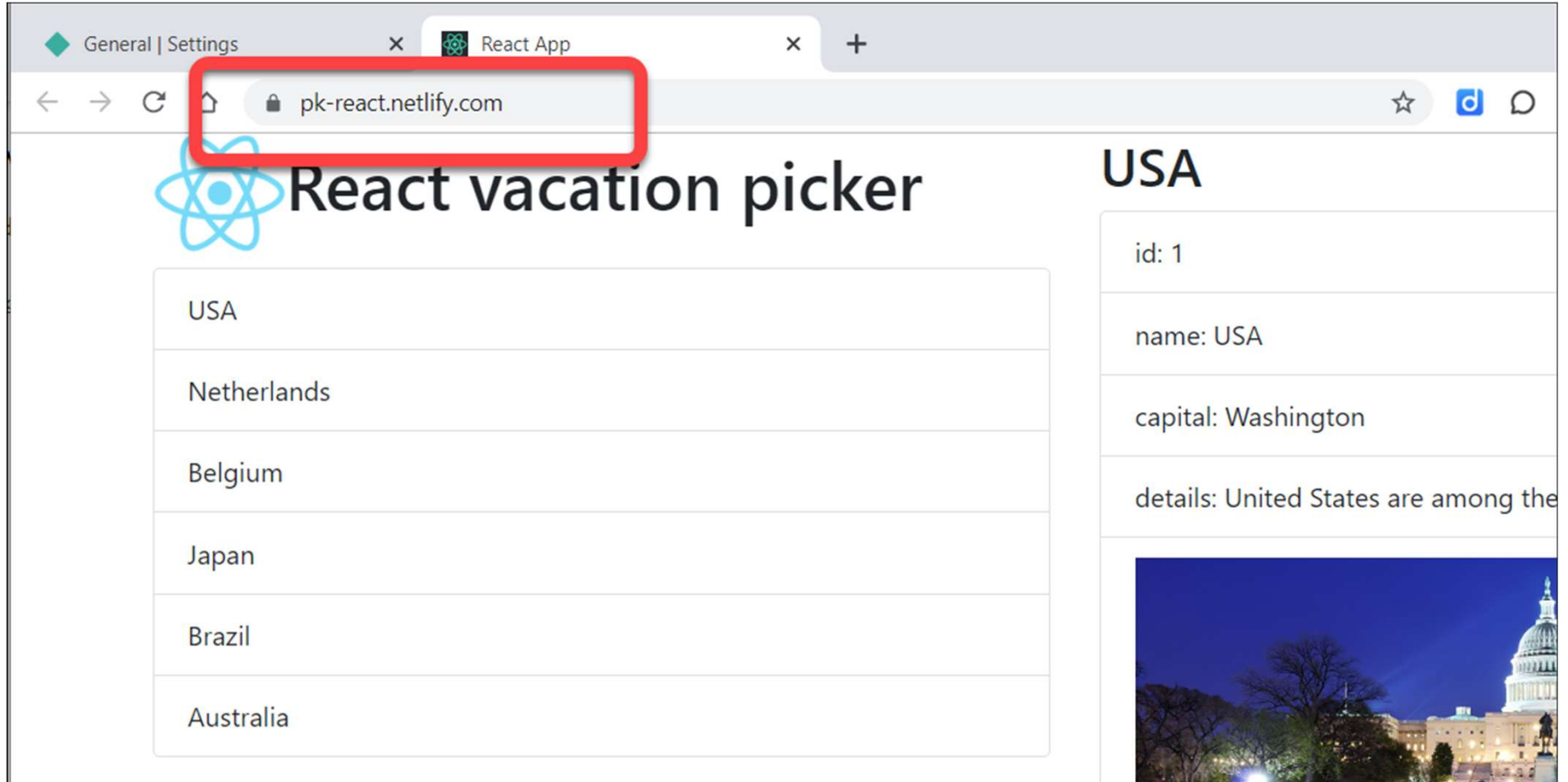
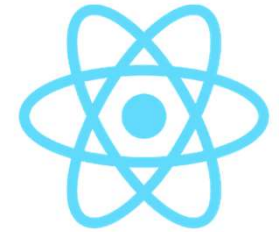
Update the name



- Via the dashboard: site details, change site name



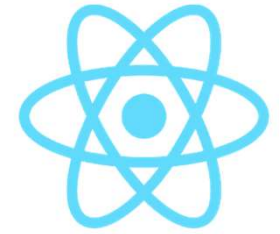
New result



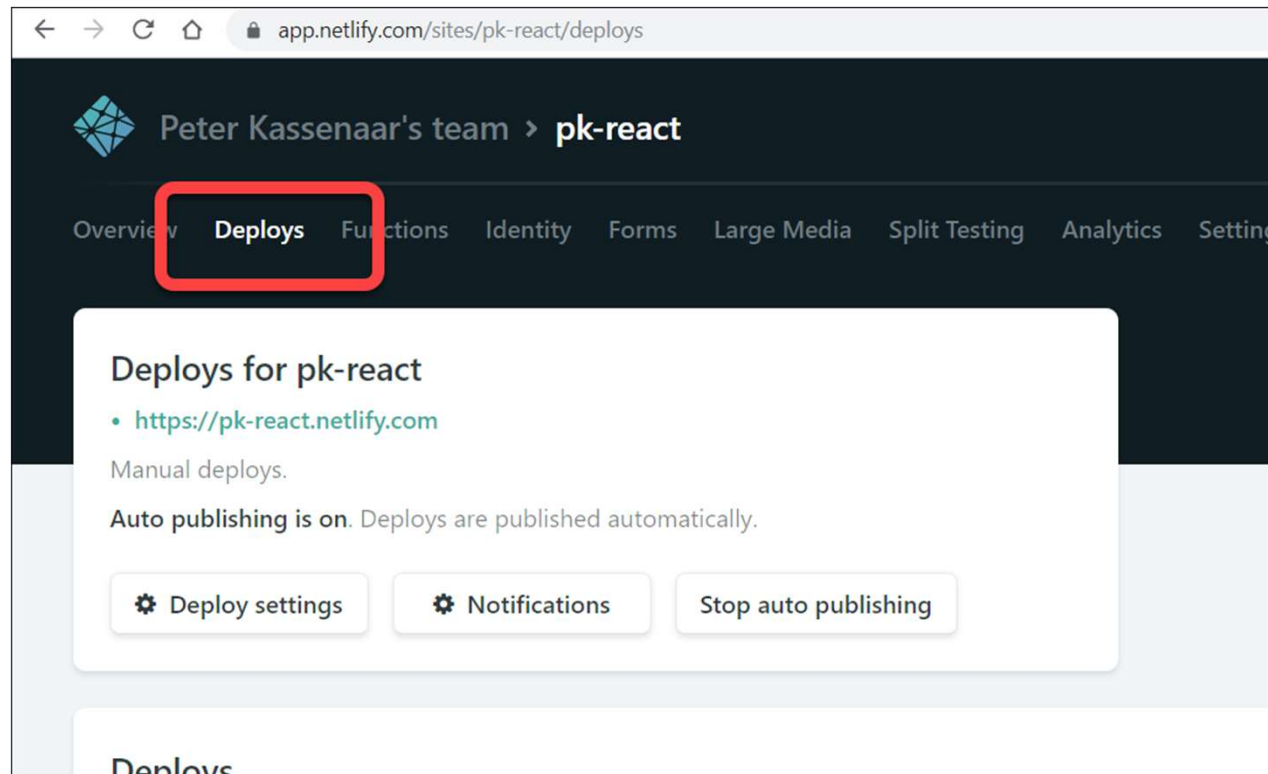
The screenshot shows a web browser with two tabs: 'General | Settings' and 'React App'. The address bar, highlighted with a red rectangle, shows the URL 'pk-react.netlify.com'. The main content area displays the 'React vacation picker' application. On the left, a list of countries is shown: USA, Netherlands, Belgium, Japan, Brazil, and Australia. On the right, a detailed view for the 'USA' is displayed, including its ID (1), name (USA), capital (Washington), and a description (United States are among the). Below the text is a photograph of the US Capitol building at night.

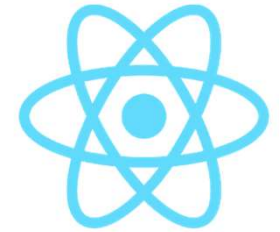
Country	id	name	capital	details
USA	1	USA	Washington	United States are among the

Re-deploy



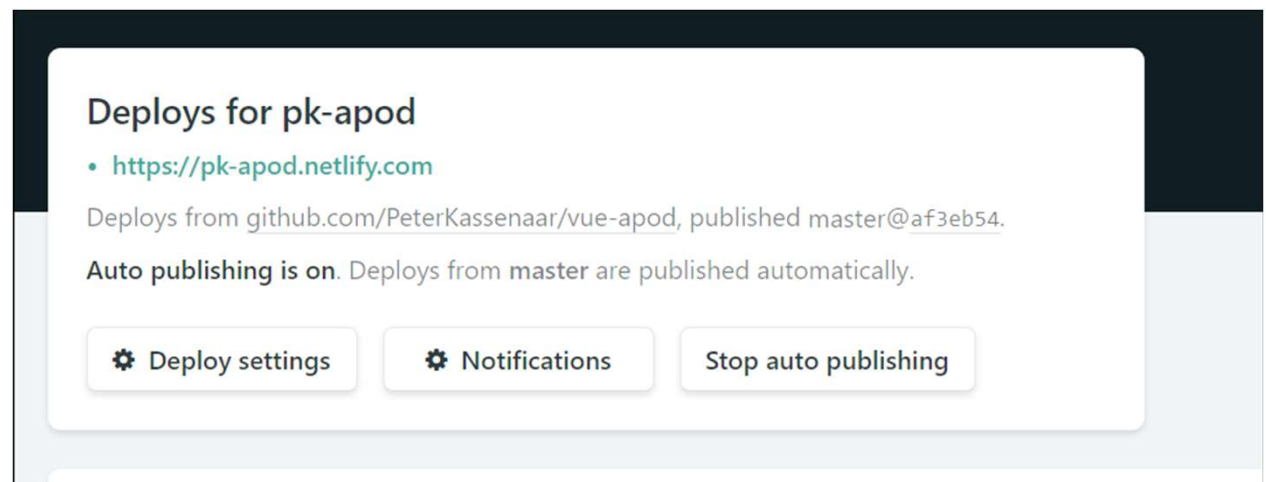
- First: update your app locally, create a new build (!)
- Go to Deploys, drag & drop the new build



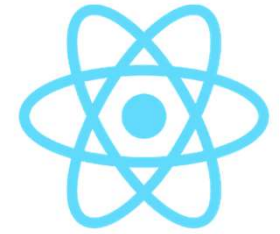


More options

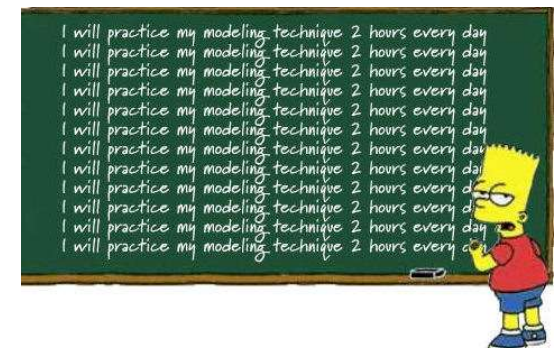
- Connect with a Github Repository
- Link with a custom domain (\$)
- Add Analytics
- Add online functions
- Delete the app
-



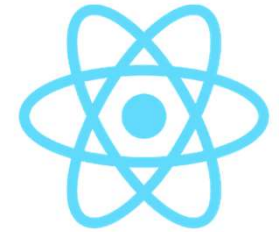
Workshop





- Use one of your own projects,
 - or start from `../500-api-call`
- Create a local build and serve from your own computer, using the `serve` package
 - Don't forget to run `npm run build` first!
- Register with Netlify and publish your project online, using the steps in this presentation
- Update the name of the project



More info



 Create React App

Docs Help GitHub 

Welcome

Getting Started

Development

Styles and Assets

Building your App

Testing

Back-End Integration

Deployment

Deployment

Advanced Usage

Support

Deployment

`npm run build` creates a `build` directory with a production build of your app. Set up your favorite HTTP server so that a visitor to your site is served `index.html`, and requests to static paths like `/static/js/main.<hash>.js` are served with the contents of the `/static/js/main.<hash>.js` file. For more information see the [production build](#) section.

Static Server

For environments using [Node](#), the easiest way to handle this would be to install [serve](#) and let it handle the rest:

```
npm install -g serve
serve -s build
```

The last command shown above will serve your static site on the port **5000**. Like many of [serve](#)'s internal settings, the port can be adjusted using the `-l` or `--listen` flags:

```
serve -s build -l 4000
```

Static Server

Other Solutions

Serving Apps with Client-Side Routing

Building for Relative Paths

Serving the Same Build from Different Paths

Customizing Environment Variables for Arbitrary Build Environments

AWS Amplify

Azure

Firebase

GitHub Pages

Step 1: Add homepage to package.json

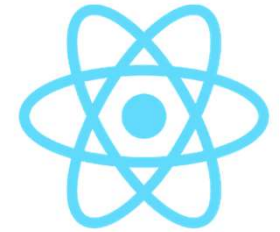
Step 2: Install gh-pages and add deploy to scripts in package.json




Step 3: Deploy the site by running npm run deploy


Step 4: For a project page, ensure your project's settings use gh-pages

<https://create-react-app.dev/docs/deployment>




Deploy to Github Pages









DEV[WRITE A POST](#)



How to deploy React App to GitHub Pages

 Ibrahim Ragab   Jun 23 Updated on Nov 03, 2019 · 2 min read

 80  19  135  [DISCUSS](#) 

**Ibrahim Ragab**
@yuribenjamin
Frontend developer @WuiltApp, Former Lawyer

[+ FOLLOW](#)

work

Front-end developer at Wuilt

location

Cairo, Egypt


education

Front-End Web Developer
Nanodegree

joined

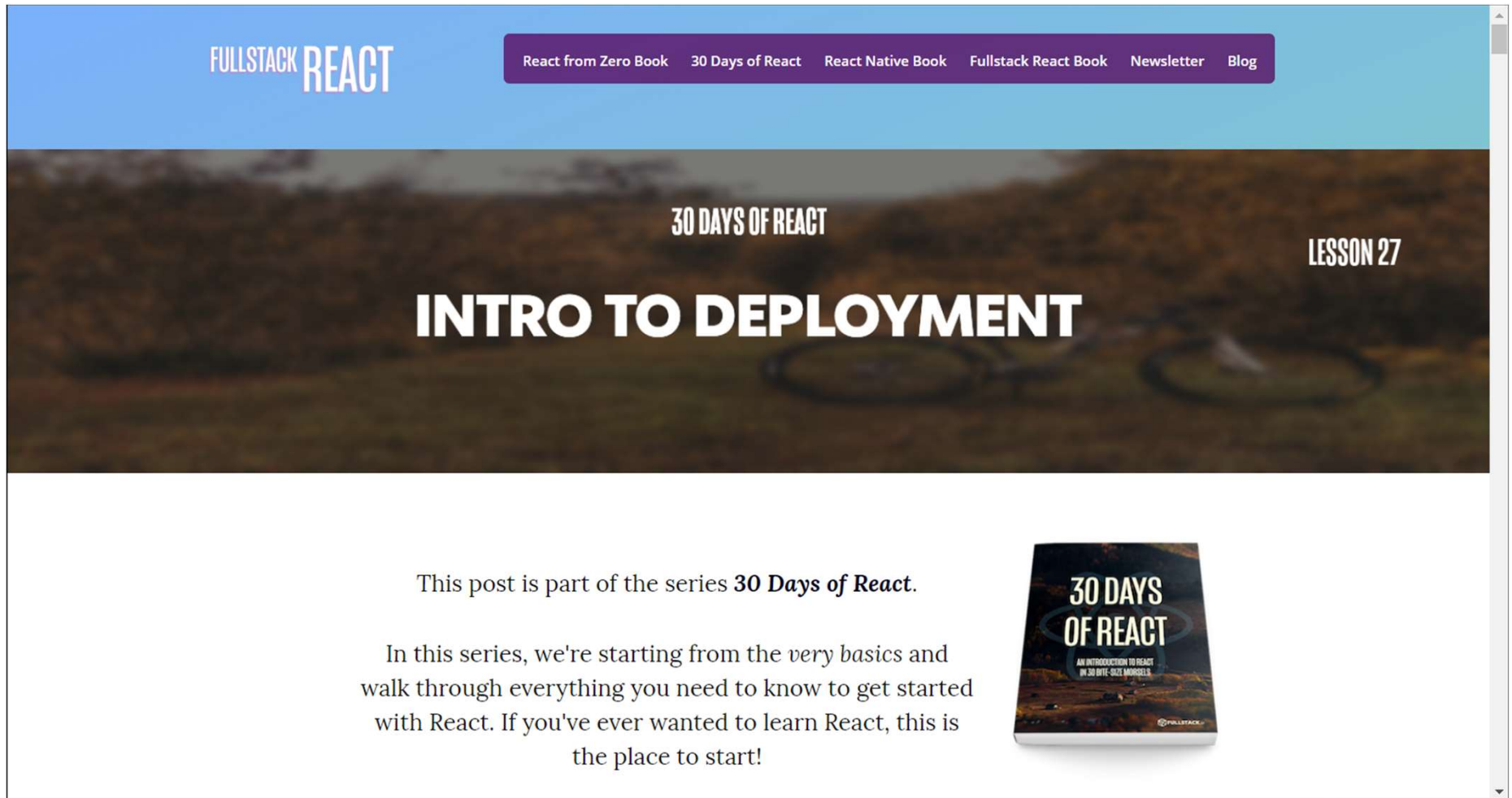
Dec 8, 2018

More from @yuribenjamin

 **Write Node server with ES6 Classes**
[#discuss](#) [#node](#) [#express](#) [#javascript](#)

<https://dev.to/yuribenjamin/how-to-deploy-react-app-in-github-pages-2a1f>

When doing more manually...

A screenshot of the Fullstack React website. The header is blue with the 'FULLSTACK REACT' logo on the left and a navigation bar on the right containing links: 'React from Zero Book', '30 Days of React', 'React Native Book', 'Fullstack React Book', 'Newsletter', and 'Blog'. The main content area has a dark, blurred background image of a field with a bicycle. It features the text '30 DAYS OF REACT' and 'LESSON 27' in the top right, and 'INTRO TO DEPLOYMENT' in large white letters in the center. Below this, on a white background, is a paragraph: 'This post is part of the series *30 Days of React*. In this series, we're starting from the *very basics* and walk through everything you need to know to get started with React. If you've ever wanted to learn React, this is the place to start!'. To the right of the text is a book cover for '30 DAYS OF REACT' with the subtitle 'AN INTRODUCTION TO REACT IN 30 DAYS - SIZE MARGARET'.

FULLSTACK REACT


React from Zero Book 30 Days of React React Native Book Fullstack React Book Newsletter Blog

30 DAYS OF REACT LESSON 27

INTRO TO DEPLOYMENT

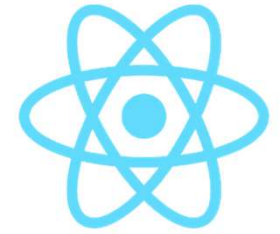
This post is part of the series *30 Days of React*.

In this series, we're starting from the *very basics* and walk through everything you need to know to get started with React. If you've ever wanted to learn React, this is the place to start!



<https://www.fullstackreact.com/30-days-of-react/day-27/>

Checkpoint



- You know how to build a production version of your app
- You can serve the project locally
- You are able to deploy the site online