<itc>

React

prep

< itc >

Agenda

- ❖ Learn how to run a webpage with **React**
- ❖ Learn how to serve webpage via a Node server
- ❖ Create folder structure for the next 2 weeks

Using our React template

Folder Structure

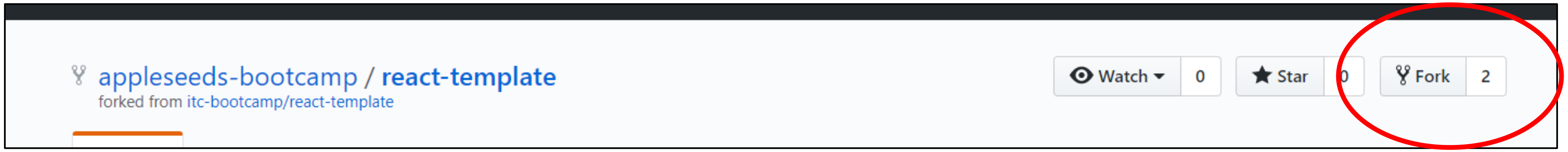
Now we will build the folder structure for the upcoming two weeks.

The full template is available at:

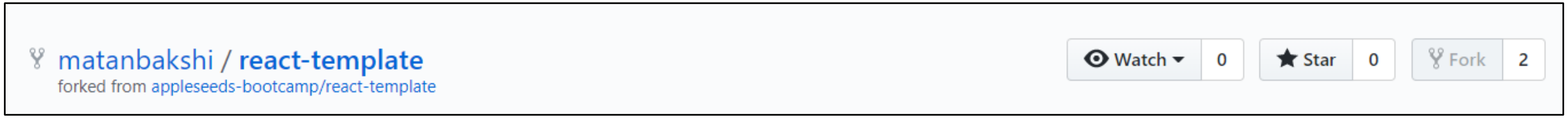
https://github.com/appleseeds-bootcamp/react-template

Using our React template

Fork the Template's Repository


appleseeds-bootcamp / react-template
forked from itc-bootcamp/react-template
Watch 0
Star 0
Fork 2

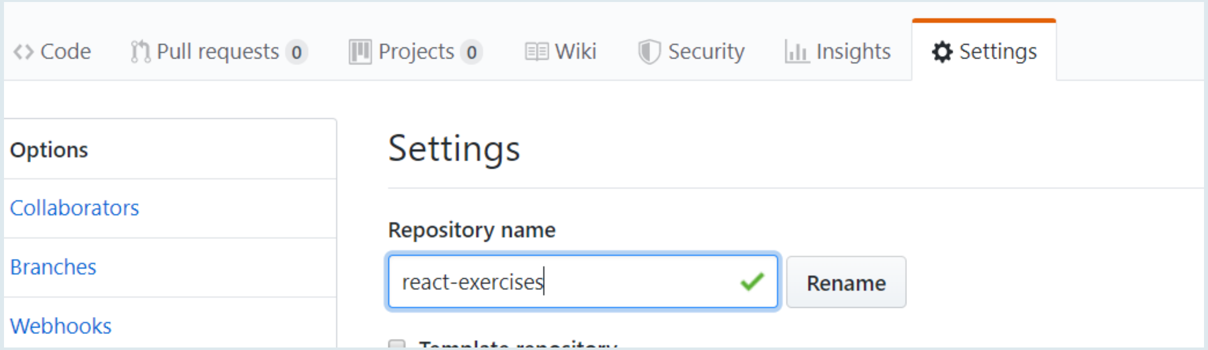
Forking the repository will create a new repository on our GitHub account which is an exact copy of the template's repository


matanbakshi / react-template
forked from appleseeds-bootcamp/react-template
Watch 0
Star 0
Fork 2

Using our React template

Rename the forked repository

On the repo’s GitHub page, click “Settings” and rename the repo to “react-exercises”



Settings

Repository name

react-exercises

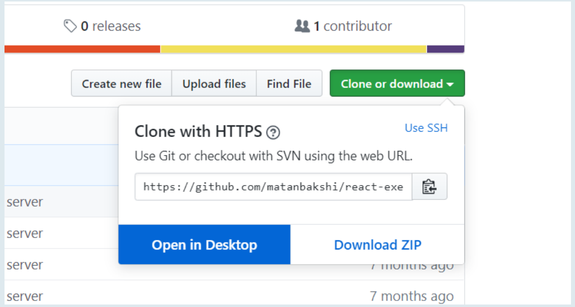
Using our React template

Clone our new repository

Clone the new repository in our /dev directory using git's clone command:

```
$ git clone https://github.com/<my-username>/react-exercises.git
```

Find the url in the repo's GitHub page.



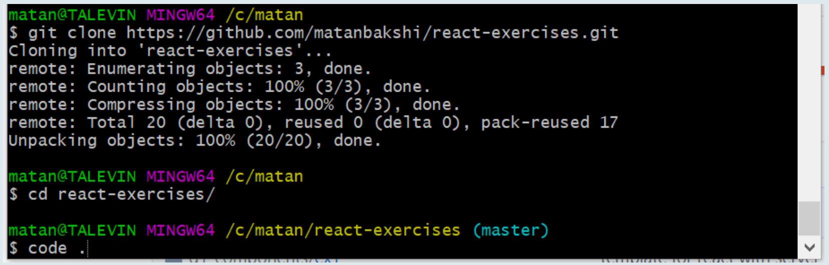
Use Git or checkout with SVN using the web URL.

```
https://github.com/matanbakshi/react-exe
```

Using our React template

Open the cloned repository with VSCode

CD into the repo's folder and open VSCode inside.

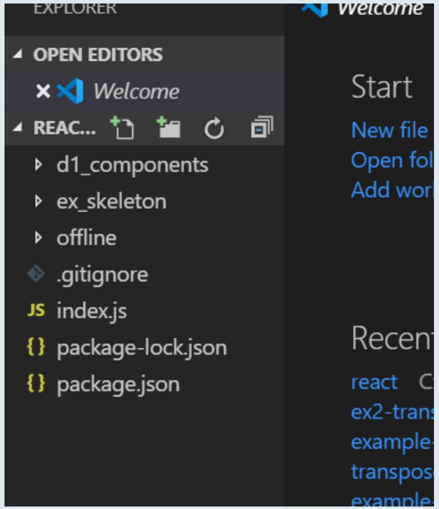


```
matan@TALEVIN MINGW64 /c/matan
$ git clone https://github.com/matanbakshi/react-exercises.git
Cloning into 'react-exercises'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 20 (delta 0), reused 0 (delta 0), pack-reused 17
Unpacking objects: 100% (20/20), done.

matan@TALEVIN MINGW64 /c/matan
$ cd react-exercises/

matan@TALEVIN MINGW64 /c/matan/react-exercises (master)
$ code .
```

Your project should look like this:



installing dependencies

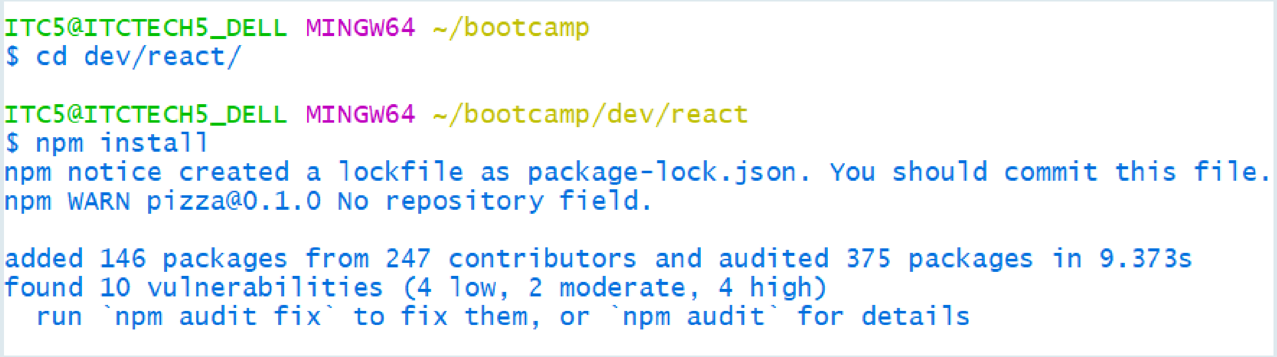
npm install

in order to install the relevant dependencies, such as express (simple server package):

open your cmd.

cd to your react exercises folder

run: npm install



```
ITC5@ITCTECH5_DELL MINGW64 ~/bootcamp
$ cd dev/react/

ITC5@ITCTECH5_DELL MINGW64 ~/bootcamp/dev/react
$ npm install
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN pizza@0.1.0 No repository field.

added 146 packages from 247 contributors and audited 375 packages in 9.373s
found 10 vulnerabilities (4 low, 2 moderate, 4 high)
  run `npm audit fix` to fix them, or `npm audit` for details
```

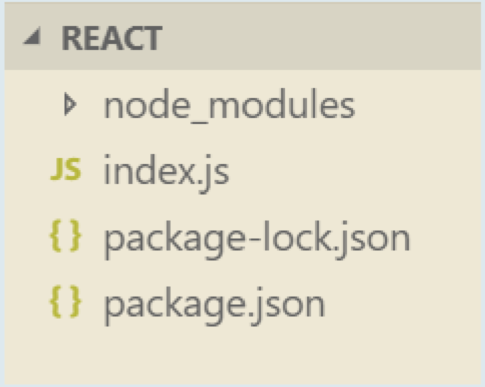
Installed!

now we have node_modules folder (contains all the packages) added

we also have another file added:

package-lock.json

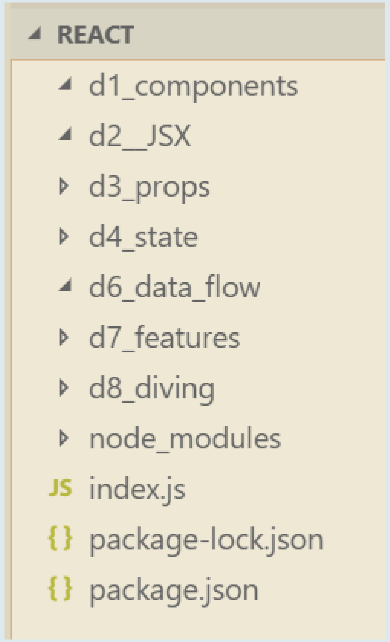
this file helps to handle the dependencies.


REACT
node_modules
index.js
package-lock.json
package.json

Folder for Each Day

We will have a folder for each **day** of react .

Let's create them:

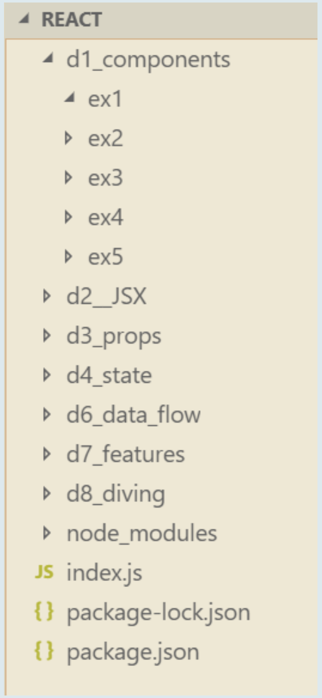


```
◢ REACT
  ◢ d1_components
  ◢ d2__JSX
  ▷ d3_props
  ▷ d4_state
  ◢ d6_data_flow
  ▷ d7_features
  ▷ d8_diving
  ▷ node_modules
  JS index.js
  { } package-lock.json
  { } package.json
```

Folder for Each Ex

We will have a folder for each **exercise** of react .

Let's create 5 ex for the first day:



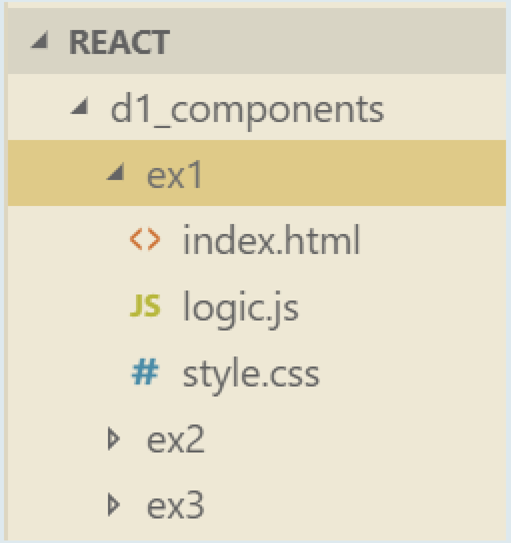
```
REACT
  d1_components
    ex1
    ex2
    ex3
    ex4
    ex5
  d2__JSX
  d3_props
  d4_state
  d6_data_flow
  d7_features
  d8_diving
  node_modules
  index.js
  package-lock.json
  package.json
```

Folder for Each Ex

inside each **exercise** we need to have the react template:

1. html file
2. js file
3. css file

The first exercise files is already in the template, use it for every exercise

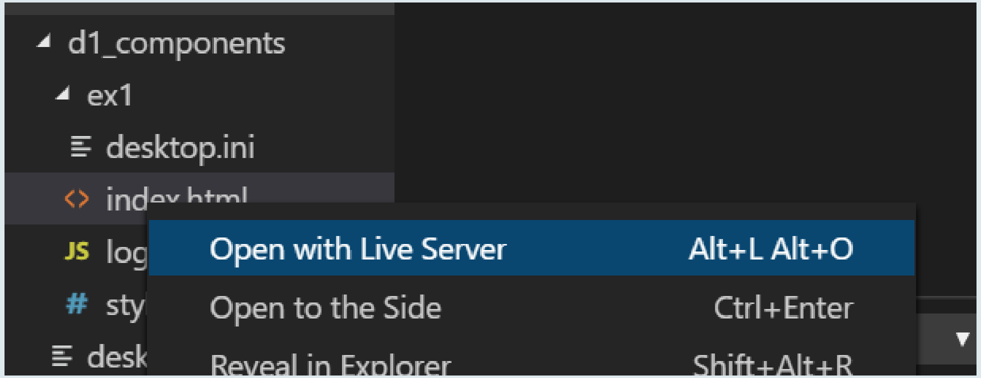


Run our app – using “Live Server”

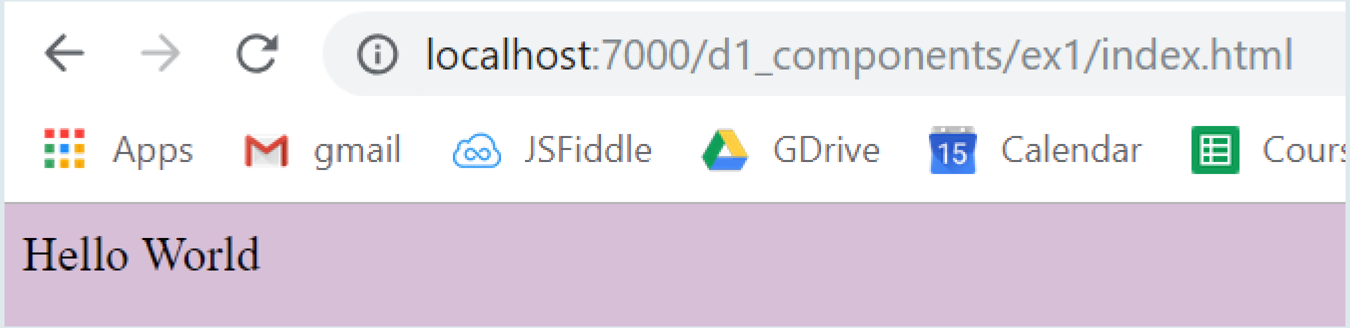
For running our app we have two options:

Using the “Live Server” VSCode plugin.

For that, you’ll need to install the “Live Server” plugin with the extensions tab inside VSCode

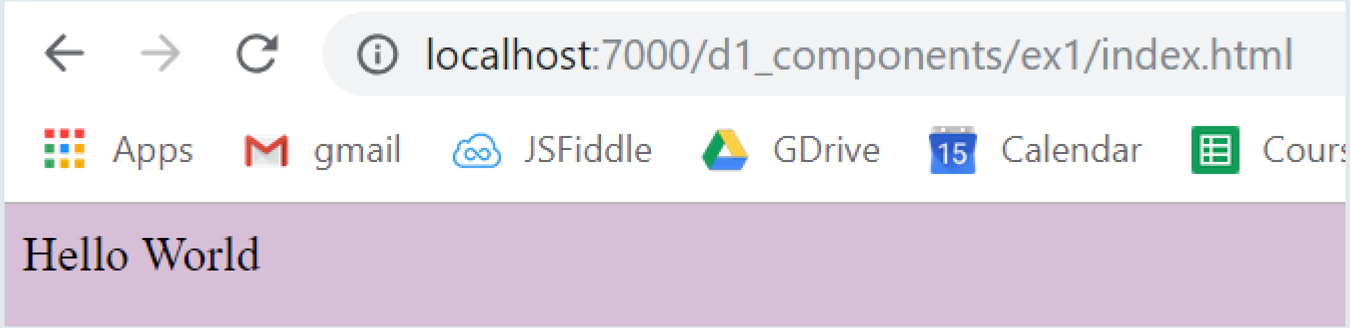

d1_components
ex1
desktop.ini
index.html
log
sty
desk
Open with Live Server Alt+L Alt+O
Open to the Side Ctrl+Enter
Reveal in Explorer Shift+Alt+R

You should get that:


localhost:7000/d1_components/ex1/index.html

Hello World

Copy that skeleton to all the other exercises:



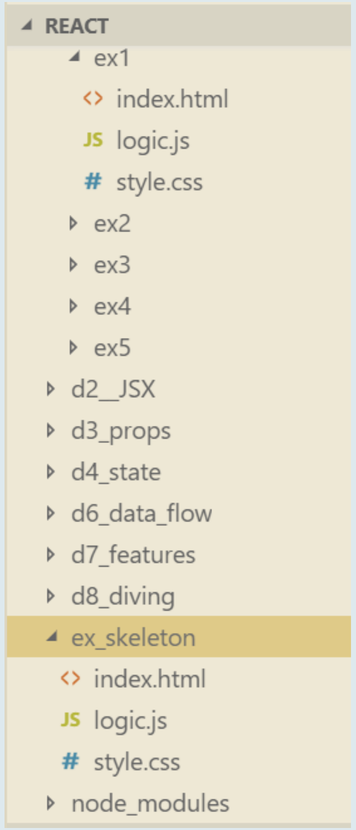
Hello World

Skeleton

We want to save one template for further use.

Create an ex_skeleton in the root folder.

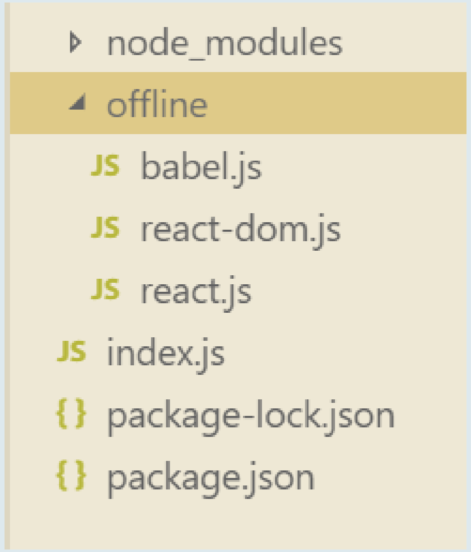
Copy paste the template into it!



Like with any library we can also work offline.

Under /offline folder you will find React's dependencies for offline use

1. react.js
2. react-dom.js
3. babel.js



```
▸ node_modules
◢ offline
    JS babel.js
    JS react-dom.js
    JS react.js
  JS index.js
  {} package-lock.json
  {} package.json
```

You will need to download the react files.

Search online, preferably in the official website.

Add references to the offline files in the skeleton.

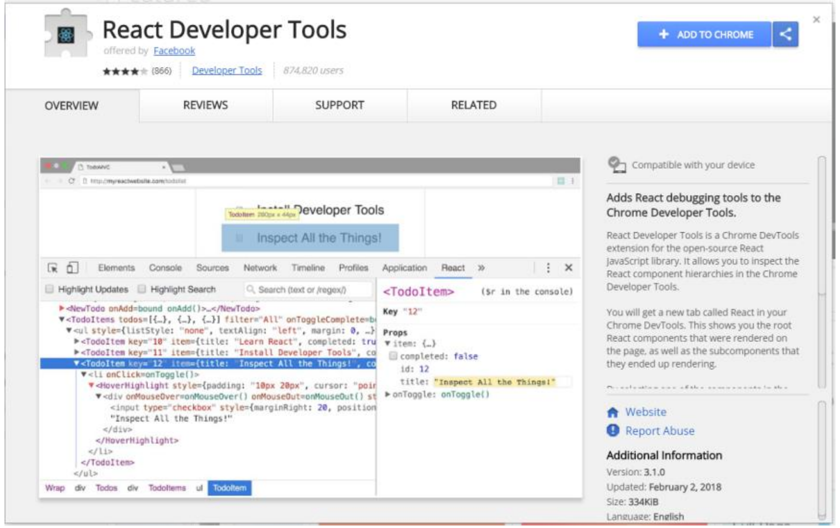
Now the head in your skeleton HTML will look like:

```
<head>
  <meta charset="utf-8" />
  <!-- cdn links -->
  <script src="https://fb.me/react-0.14.0.js"></script>
  <script src="https://fb.me/react-dom-0.14.0.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/babel-
    core/5.8.25/browser.min.js"></script>
  <!-- offline links -->
  <script src="/offline/react.js"></script>
  <script src="/offline/react-dom.js"></script>
  <script src="/offline/babel.js"></script>
  <!-- css file -->
  <link rel="stylesheet" href="./style.css">
</head>
```

Install React Dev Tools

Adds React debugging tools to the Chrome Developer Tools.

Install here.



React Developer Tools

offered by Facebook

★★★★★ (866) | Developer Tools | 874,820 users

+ ADD TO CHROME

OVERVIEW | REVIEWS | SUPPORT | RELATED

Install Developer Tools

Inspect All the Things!

Elements Console Sources Network Timeline Profiles Application React »

Highlight Updates | Highlight Search | Search (text or /regex/)

```
▶<NewTodo onAdd=bound onAdd()>…</NewTodo>
▼<TodoItems todos=[{…}, {…}, {…}] filter="All" onToggleComplete=b
  ▼<ul style={listStyle: "none", textAlign: "left", margin: 0, …}
    ▶<TodoItem key="10" item={title: "Learn React", completed: tru
    ▶<TodoItem key="11" item={title: "Install Developer Tools", co
    ▼<TodoItem key="12" item={title: "Inspect All the Things!", co
      ▼<li onClick=onToggle()>
        ▼<HoverHighlight style={padding: "10px 20px", cursor: "poir
          ▼<div onMouseOver=onMouseOver() onMouseOut=onMouseOut() st
              <input type="checkbox" style={marginRight: 20, position
              "Inspect All the Things!"
            </div>
          </HoverHighlight>
        </li>
      </TodoItem>
    </ul>
```

Wrap div Todos div TodoItems ul TodoItem

```
<TodoItem>      ($r in the console)
Key "12"
Props
▼item: {…}
  completed: false
  id: 12
  title: "Inspect All the Things!"
▶onToggle: onToggle()
```

Compatible with your device

Adds React debugging tools to the Chrome Developer Tools.

React Developer Tools is a Chrome DevTools extension for the open-source React JavaScript library. It allows you to inspect the React component hierarchies in the Chrome Developer Tools.

You will get a new tab called React in your Chrome DevTools. This shows you the root React components that were rendered on the page, as well as the subcomponents that they ended up rendering.

Website

Report Abuse

Additional Information

Version: 3.1.0

Updated: February 2, 2018

Size: 334KiB

Language: English

`the end.`