Five Minute React

Video Updates

Hello and welcome to Five Minute React! This document is meant to alert you to some changes and fixes that have happened to the course since the videos were recorded. The full-text versions of the tutorials have been updated, but the videos now contain some out-of-date information. This document will help you quickly get around any discrepancies in the videos without having to hunt through the text.

Only tutorials with significant changes are listed.

Tutorial 2

Some Sublime Text package names have changed slightly. Install these packages:

- Babel (Syntax definitions for ES6 JavaScript with React JSX Extensions)
- Sass
- SublimeLinter
- SublimeLinter-ESLint

Tutorial 3

Installation of NVM remains the same as in the video on both platforms, but the versions of Node used in the video are out of date. Please work only with Node version 8.11.1 – even if by the time you get to this, that version is also out of date. 8.11.1 is guaranteed to work with this series.

Tutorial 4

Yarn now asks for an additional parameter on setup: `private`. It doesn't really matter whether you choose private or public.

Tutorial 6

Slight changes to version numbers, and MongoDB now requires a couple of extra clicks to download the Community Server (which is the one you want).

Tutorial 10

Express.js v4.16.x and the associated express-generator module make significant changes to the generated `app.js` file. Please make sure you are using Express v4.15.5 and Express-Generator v4.15.5. You can do this by typing `yarn global add express-generator@4.15.5`

Tutorial 16

Make sure to version-lock Mongoose by using `yarn add mongoose@5.04`

Tutorial 17

REMOVE `password: string` from your user model. It's unnecessary when working with Passport.

Your finished `package.json`, before running yarn, should look like this:

```
{
  "name": "musiclist",
  "version": "0.0.0",
  "private": true,
  "scripts": {
    "start": "node ./bin/www"
  },
  "dependencies": {
    "body-parser": "~1.17.1",
    "cookie-parser": "~1.4.3",
    "debug": "~2.6.3",
    "ejs": "~2.5.6",
    "express": "~4.15.2",
    "express-session": "1.10.1",
    "mongoose": "5.0.4",
    "morgan": "~1.8.1",
    "passport": "0.3.2",
    "passport-local": "1.0.0",
    "passport-local-mongoose": "4.0.0",
    "serve-favicon": "~2.4.2"
  }
}
```

Tutorial 18

Version lock jest by using `yarn add jest@19.0.2 --dev`.

Version lock supertest by using `yarn add supertest@3.0.0 --dev`.

You can ignore everything about storing plain-text passwords in the database since we're not, in fact, doing that.

Use this command in Mongo instead: `db.users.insert({ "username": "administrator", firstName": "YOUR FIRST NAME", "lastName": "YOUR LAST NAME" });` (where, of course, you use your actual first and last names).

Tutorial 20

You no longer need to open `/models/user.js` and change the password line, because that line no longer exists.

Tutorial 21

Use `npm info "eslint-config-airbnb@15.0.1" peerDependencies`

... this will give you a peerDependencies object identical to the one in the tutorial / video, but make sure to version-lock by changing the yarn add line to the following: `yarn add --dev eslint-config-airbnb@15.0.1 eslint@^3.19.0 eslint-plugin-jsx-a11y@^5.0.1 eslint-plugin-import@^2.2.0 eslint-plugin-react@^7.0.1`

When fixing the linting errors in `app.js`, do not uncomment line 25 (this means you'll have to live with the error on line 3). It'll break your app. Of course, we re-comment it in video 25, so even if you do uncomment it, it's not the end of the world!

Tutorial 22

Globally add ESLint and its dependencies with this line: `yarn global add eslint-config-airbnb@15.0.1 eslint@^3.19.0 eslint-plugin-jsx-a11y@^5.0.1 eslint-plugin-import@^2.2.0 eslint-plugin-react@^7.0.1`

Add `eslint-plugin-jest` with these two lines: `yarn add --dev eslint-plugin-jest@20.0.3` and `yarn global add eslint-plugin-jest@20.0.3`

Again, we no longer have a `password` line in our code, so ignore that.

Because we didn't change the favicon import in `app.js`, it's now throwing a lint error, so fix that by changing it to this code: `const favicon = require('serve-favicon'); // eslint-disable-line`

Tutorial 23

Near the end of the tutorial, we install a whole bunch of stuff. Use this command to version-lock all of it: `yarn add react@16.1.1 react-dom@16.1.1 webpack@3.4.1 babel-loader@7.0.0 babel-core@6.24.1 babel-jest@20.0.3 babel-preset-react@6.24.1 sass-loader@6.0.6 node-sass@4.5.3`

Tutorial 24

We're moving from using `babel-preset-es2017` to `babel-preset-env`, which contains transpilers for ES2015, ES2016, ES2017. So use `yarn add babel-preset-env@1.6.1`

Tutorial 25

You can ignore the stuff about re-commenting favicon.

Tutorial 27

Version lock the two modules you're installing by using this line: `yarn add --dev webpack-dev-server@2.4.5 react-hot-loader@3.0.0`

Tutorial 28

Do not actually add `react-hot-loader@next`. We've version-locked now and no longer need to do that.

Also, you can ignore the "fix" for `webpack.config.js` at the end - we already fixed it in a previous tutorial.

Tutorial 31

Version-lock `react-router-dom` by installing it with this line: `yarn add react-router-dom@4.1.1`

Tutorial 32

Version-lock your Webpack middleware by installing it with this line: `yarn add webpack-dev-middleware@1.11.0 webpack-hot-middleware@2.18.0`

Also, I've added an additional step to help make hot reloading more robust. Open `/src/index.jsx` and replace lines 17 to 21 (the module.hot stuff) with the following:

```
if (module && module.hot) {
  module.hot.accept();
}
```

Tutorial 34

There's a large bunch of module additions at the beginning of this tutorial. Version-lock them by using this line: `yarn add reactstrap@4.8.0 react-addons-transition-group@15.6.0 react-addons-css-transition-group@15.6.0 bootstrap@4.0.0-alpha.6 extract-text-webpack-plugin@2.1.2 css-loader@0.28.4 style-loader@0.18.2`

Tutorial 36

Version-lock the two modules installed at the beginning of the tutorial with this line: `yarn add sass-loader@6.0.6 node-sass@4.5.3`

Tutorial 37

You can ignore the bit about storing user passwords in plaintext (and deleting the associated line of code) because we never actually did that. Woo!

Tutorial 39

Version-lock Redux and its associated modules by installing them with this line: `yarn add redux@3.7.2 react-redux@5.0.5 redux-devtools@3.4.0`

Tutorial 40

The syntax for hot loading at the end of `index.jsx` no longer needs to change. Just leave it alone!

Tutorial 42

Version-lock your Redux tools by installing them with this line: `yarn add --dev redux-logger@3.0.6 redux-devtools-log-monitor@1.3.0 redux-devtools-dock-monitor@1.1.2`

The groundwork we're laying for hot-reloading reducers will currently throw an error in your browser console the first time your hot-reloader fires. You can safely ignore this for now.

Tutorial 43

You can ignore everything about installing `babel-preset-es2015`, and removing `babel-preset-es2017`. We're no longer using those, and what we are using works without making any tweaks.

You no longer need to edit .babelrc

Tutorial 44

We no longer need to change `webpack.config.js` to `webpack.config.babel.js` or update any scripts to deal with that.

We also don’t need to worry about the trailing comma in the Webpack config anymore, nor does our development server die on us, so you don’t need to install and set up `babel-register` .

Also, for some reason I have us recreate the `start-prod` script from the previous tutorial. You … don’t need to do that.

Tutorial 45

This tutorial starts by going over the `babel-register` stuff we ignored in the last one. You can ignore it here, too.

Version lock `whatwg-fetch` by using this command to install it: `yarn add whatwg-fetch@2.0.3`

Tutorial 46

Using an unconfigured `babel-plugin-env` instead of `babel-plugin-es2017` breaks `async / await` support (isn’t coding fun?) but it’s easily fixed with a quick tweak to your `.babelrc` file. Make it look like this:

```
{
  "presets": [
    "react",
    [
      "env", {
        "targets": {
          "browsers": ["last 2 Chrome versions"]
        }
      }
    ],
  ],
}
```

Tutorial 51

Version-lock redux thunk using `yarn add redux-thunk@2.2.0`

Tutorial 58

Version-lock `availity-reactstrap-validation` with this command: `yarn add availity-reactstrap-validation@1.4.3`

Tutorial 62

Version-lock the Mailgun module by installing it with this code: `yarn add mailgun-js@0.13.1`

Tutorial 63

Our `app.js` no longer has `require('babel-register')` in it, so you can ignore that.

Add `const appConfig = require('./config.js');` at the top of the file, rather than under line 1, so it stays in alphabetical order.

Tutorial 64.5

You don't need to add `babel-preset-es2017`. We're using `babel-preset-env` now, and it works fine, we just need to tweak the code in our webpack config file. For both the JSX and JS "query" lines, use this code: `presets: ['env']`,

Tutorial 67

Version lock disconnect with this code: `yarn add disconnect@1.2.1`

Tutorial 68

TheDevTools logger has a bug that makes it very unhappy if you use an array for `initialState` in your reducer. To fix this bug, head your your terminal or command prompt, kill your server, and type: `yarn add --dev redux-devtools-dock-monitor@1.1.3 redux-devtools-log-monitor@1.4.0` ... Once that runs, restart your server.

Tutorial 70

Do not delete line 8. That's referring to the password line, which we removed long ago.

Tutorial 71

Due to the way newer versions of MongoDB work, some code in `/api/albums.js` has changed. Your `saveArtists` function should look like this:

```
// Check each artist in an array to see if it exists and if not, save it
const saveArtists = async (artists) => {
  const formattedArtists = artists.map((artist) => {
    const newArtist = Object.assign({ discogsId: artist.id }, artist);
    return newArtist;
  });
  try {
    const result = await new Promise((resolve) => {
      Artist.insertMany(
        formattedArtists,
        { ordered: false },
        (error) => {
          if (error && error.code !== 11000) {
            resolve(false);
          }
          resolve(true);
        },
      );
    });
    return result;
  } catch (error) {
    if (error.code !== 11000) {
      return false;
    }
    return true;
  }
};
```

Tutorial 76

Huge chunks of this tutorial are no longer relevant, but some of the changes to `package.json` are still useful. I strongly recommend just biting the bullet and checking out the full-text version.

Tutorial 77

There are a bunch more handleKeypress methods to delete. The list of files is:

- `/src/components/account/ChangePasswordPage.jsx`
- `/src/components/account/LoginPage.jsx`
- `/src/components/account/RegisterPage.jsx`
- `/src/components/account/ResetPasswordPage.jsx`
- `/src/albums/AlbumsPage.jsx`
- `/src/components/artists/ArtistsPage.jsx`

Tutorial 78

Version lock your compression files with this code: `yarn add compression-webpack-plugin@1.0.1 compression@1.7.1`

Tutorial 79

Version lock your modules with these commands:

- `yarn add helmet@3.9.0`
- `yarn add express-rate-limit@2.9.0`
- `yarn add dompurify@1.0.2 jsdom@11.5.1`

Running `snyk wizard` comes with some issues now – we don't want to upgrade our Express version, for example. So we're not going to run the wizard anymore. If you're writing actual production code, you'll need to close up those vulnerabilities and deal with any errors that process introduces … but since this is a tutorial, I think it's more valuable to wrap things up than to introduce extra "fixing what broke" steps. So don't run `snyk wizard` at all.

You're Done!