A Way To Upgrade To React

Upgrade modules to React!

By Vaishak Kallore
/vaikoovery

Why did you choose React.js?

To be frank, I had never heard about *React.js* till one of my friends, visited me back in December, 2015.

He had given an intro to it, then I tried out few code and liked the way it worked, though I didn't understand it much, I liked the props and state ;) & it was v0.14.x then.

How did you start the work on ‘real’ apps?

I wanted to start with the work on the **real project** and decided to start with a module in our existing website which was in pipeline to develop.

React.js Scripting?

Being a traditional developer started writing the script under '*assets/js*' directory.

Including React files in website?

I had started with including the '*jsx*' files directly in the browser by loading the '*react* & *react-dom*' CDN files.

Did it go well?

NOOO! I had failed, when I wanted to add new packages. I started getting pain in *** :)

What did you do then?

I told my friend the way I worked, he has asked me to change my approach by using '*webpack*'.

Well, I had gone through the docs and started making the config file to make bundles.

Was it easy to get started with webpack?

Again another NOOO!

I had to try & try & try & try (...) to make it working atlast.

How did you include the bundled files on your website?

As it was the only page which was using '*React.js*', I had included the bundled file directly.

Did you find it easy to go forward?

NOOO!

As all know, I had to face the caching issues when I updated the code.

How did you handle the file caching issue?

I had spent some more time on '*webpack*' and came across hashing the file names on every '*webpack*' execution.

I had to spend time again on finding out the solution for removing old files as well (*Please note, its bad, but I was & still committing the bundled files! Lack of DevOps?*).

I had used '*clean-webpack-plugin*' package to do this for me.

When it generates random file names, how do you include those while executing?

Came across the plugin '*assets-webpack-plugin*' and did a work around with that.

It contains the latest hashed file names - JS & CSS in a json file which is specified in the '*webpack*' config.

Sample *json* output


```
{
 "about-us": {
   "js": "about-us.44b4e5f89bdd84c18b20.bundle.js"
 },
 "commons": {
   "js": "commons.44b4e5f89bdd84c18b20.chunk.js",
   "css": "commons.44b4e5f89bdd84c18b20.css"
 },
 "contact-us": {
   "js": "contact-us.44b4e5f89bdd84c18b20.bundle.js"
 }
}
```

Making it dynamic...

Then I applied own logics to load the related JS & CSS files on the page, by hard coding the *entry points* in server coding, reading the '*json*' file for required entry point & loading those required CSS & JS into the page. Also set a flag on server side to load other required JS & CSS on other pages.

With this logic, I could upgrade other modules on our website to ***React Magic***

To React.js v15

In few months '*React.js*' released '*v15.x.x*' and I thought I must do an upgrade and started working on it.

I also had to upgrade the '*webpack*' to '*v2.x.x*', with few API changes.

Now, how can I start with React 16?

You can consider using '*webpack*' '*v4.x.x*'. (which has many feature, compared to old)

Have tried to demonstrate the usage by creating a project using PHP, React16 & webpack v4.x.x

https://github.com/vkallore/a-way-to-upgrade-to-react

Questions?

Thank You
&
Thank You React :)