Handwritten Notes

"PARCEL IS A BEAST"

Notes on Parcel ⟶▷

🔖 Save for later

Author : Ashrayakk

In React, to get external functionalities, we use "BUNDLERS" :-

a) Webpack is a bundler
b) vite
c) parcel

These are alternatives

In create-react-app, the bundler used is "webpack"

Most bundlers do the same job.

Bundlers are packages. If we want to use a package in our code, we have to use a 'package manager'.

We use a 'package manager' known as npm or yarn.

npm :- No Problem Man

'npm' doesn't mean node package manager but everything else.

npm init (Create a package.json file)

We use npm because we want a lot of packages in our project / react app.

[npm init -y] - will skip lot of options.

npm install -D parcel → devdependency - becoz we want it in our developer machine.

→ parcel is one of the dependency

Then, we'll get package-lock.json.

[Caret & tidle sign → read.]

our project will automatically update if we use caret sign. (^)

```
"devDependencies" : {
    "parcel" : "^2.8.2"    → caret
}
```

package-lock.json

will tell you which exact version of the library you are using.

Common issue

?) "Something is working on my localhost / mymachin but not works in production". Why?

A) · package-lock file tells the exact version of the library we are using. Suppose, for dev dependency we're using 2.8.4 version & is right in package-lock.json file.

But in 'package.json' file, it will be like "2.8.2"

'package-lock' file is an file that locks the version

"node_modules"

→ which gets installed is like a database for the npm.

→ This is where the Super powers comes from.

→ Our app has dependency on 'parcel'

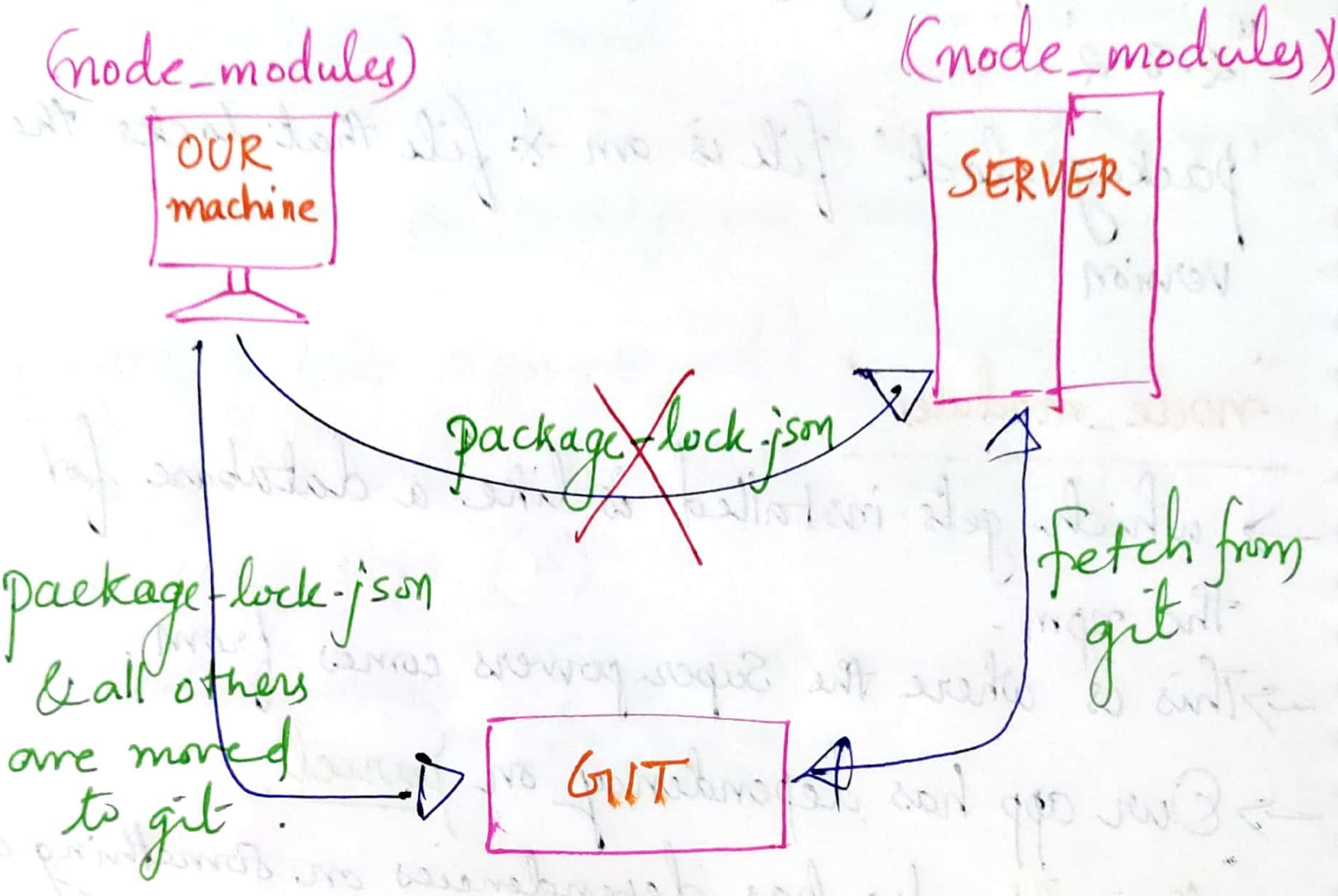
→ 'parcel' also has dependencies on something else

All these dependencies/superpowers are in 'node_modules'.

?) We don't put 'node_modules' into git. why?

→ Because, our package-lock.json file have Sufficient information to recreate node_modules.

→ package-lock.json file keep & maintain the version of everything in node_modules.



→ 'node_modules' in our machine can be regenerated in server using the package-lock.json file.

Previously, we used CDN links to get react into our app. This is not a good way.
We need to keep react in our node-modules.

```
npm install react
```

npm i react-dom

To ignite our app.

```
npx parcel index.html
```

npx ▷ means 'execute using npm'

index.html ▷ is the entry point

Click 'Enter'

Then, a miniserver is created for us:
like localhost:1234
Parcel given a server to us.

"Parcel" ignited our app.

As we removed CDN links, we don't have react in our app.
So, we want to import it into our app.
For that we use the keyword "import"

Never touch 'node_modules' & 'package-lock.json'

Normal js browser don't know "import".
So, it shows error.

```
import React from "react";
import ReactDOM from "react-dom"/client";
```

As we got an error, we have to specify to the browser that 'we are not using a normal Script tag, but a module'.

```
<script type="module" src="App.js"> </script>
```

We cannot import & export scripts inside a tag. Modules can import and export.

Note :

* Hot Module Reload (HMR)

→ means that parcel will keep a track of all the files which you're updating.

* How HMR works ?

→ There is File Watcher Algorithms (written in C++). It keeps track of all the files which are changing realtime & it tells the server to reload.

→ These all are done by "PARCEL".

* There'll be a folder called .parcel-cache which will be there automatically.

In our project, parcel needs some space. So, it creates .parcel-cache.

* 'dist' folder keeps the files minified for us.

When we run command:

```
npx parcel index.html
```

This will creates a faster development version of our project & serves it on the server.

When I tell parcel to make a production build:

```
npx parcel build index.html
```

It creates a lot of things, minify your file.

And "parcel will build all the production files to the dist folder"

* What takes a lot of time to load in a website?

A):- Media - Images

Parcel does image optimization also.

* Parcel also does 'Caching while development'.

* Parcel also takes care of your older version of browser

Compactible with older version of browsers'.

Sometimes we need to test our app on https becoz something only works on https.
Parcel gives us a functionality that we can just build our app on https. on dev machine.

```
npx parcel index.html --https
```

* Should put the .parcel-cache in gitignore

↳ because, anything which can be auto-generated should be put inside gitignore.

* Parcel ~~does~~ uses:

Consistent Hashing Algorithms

* Parcel is 'Zero Config'.

Parcel features in a glance :

-> HMR - Hot Module Reloading

-> File Watcher algorithm - C++

-> Bundling

-> minify code

-> Cleaning our code

-> Dev and production build

-> Super fast build algorithm

-> Image Optimization

-> Caching while development

-> Compression

-> Compactible with older browser version

-> HTTPS on dev

-> port Number

-> Consistent Hashing Algorithm

-> Zero Config

-> Tree Shaking