

Chapitre 2 : Introduction au React JS

Table des matières

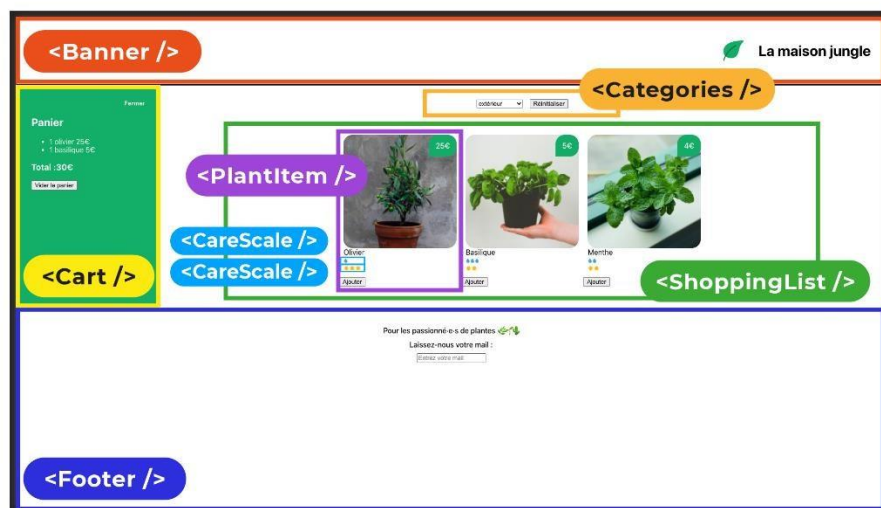
| | | |
|-------|----------------------------------|---|
| I. | Présentation de React..... | 2 |
| II. | Node JS..... | 2 |
| III. | NPM..... | 3 |
| a) | La commande npm..... | 3 |
| b) | La commande npx..... | 4 |
| IV. | L'outil create react app | 4 |
| V. | Structure d'un projet React..... | 4 |
| VI. | Première application..... | 4 |
| VII. | Les extensions de VS code..... | 7 |
| VIII. | Les extensions de chrome..... | 7 |
| IX. | Exercice d'application :..... | 8 |

I. Présentation de React

React (React JS ou ReactJS) est une bibliothèque JavaScript libre développée par Facebook depuis 2013.

React est une bibliothèque JavaScript permettant de construire des interfaces utilisateur **composables**. Elle encourage la création de composants d'interface utilisateur réutilisables, qui présentent des données qui évoluent dans le temps. Beaucoup de gens utilisent React comme le V de MVC (Modèle-Vue-Contrôleur).

Un composant est un élément qui contribue à la construction d'une interface utilisateur. Il peut s'agir d'une zone de texte, d'un bouton, d'un formulaire entier, d'un groupe d'autres composants, etc. Même l'interface utilisateur de l'application entière est un composant.



II. Node JS

Si tu codes déjà côté client, tu utilises forcément JavaScript. **Pourquoi ?** Parce que tous les navigateurs sont équipés d'un moteur JavaScript qui va permettre de traduire notre code JavaScript en code machine.

Un jour, Ryan Dahl qui a eu la brillante idée de prendre le moteur JavaScript V8, c'est celui qu'on trouve dans le navigateur Chrome, et de l'utiliser en dehors du navigateur. Il a créé la plateforme Node.js !

NodeJS est un environnement d'exécution gratuit permettant d'utiliser le JavaScript côté serveur.

Ce qu'on peut faire avec Node.JS :

- Node.js peut générer du contenu de page dynamique.
- Node.js peut créer, ouvrir, lire, écrire, supprimer et fermer des fichiers sur le serveur.
- Node.js peut collecter des données de formulaire
- Node.js peut ajouter, supprimer et modifier des données dans votre base de données.

Parmi les avantages de Node.Js qui nous s'intéresse dans ce cours est qu'il possède un système de gestion de paquets intégrés (NPM).

III. NPM

NPM est l'acronyme de Node Package Manager comme son nom l'indique est le "package manager" ou le gestionnaire de package officiel de l'univers JavaScript (*frontend / backend*). Il est installé automatiquement lors de l'installation de NodeJS.

a) La commande npm

La commande npm est utilisé pour installer et mettre à jour les paquets déjà installés dans votre projet.

- 1) pour déterminer la version de npm : **npm -v**
- 2) Mettre à jour NPM vers la dernière version : **npm install npm@latest -g**
- 3) installer un package : **npm install package_name** ou **npm i package_name**
- 4) Mettre à jour un package déjà installe : **npm update package_name**
- 5) Exécuter un package déjà installe : **npm run package_name**
- 6) vider le cache : **npm cache clean --force**
- 7) désinstaller un package : **npm -g uninstall --save**

Si vous utilisez la commande "npm mon-paquet", cela n'exécutera le paquet que s'il a été installé globalement. Vous pouvez donc avoir un message d'erreur pour un paquet installé localement. La solution pour exécuter un paquet local est de saisir le chemin entier vers le paquet.

b) La commande npx

npx est l'acronyme de Node Package eXecute. C'est un outil spécialement conçu pour l'exécution des paquets. Lorsque vous lancez l'exécution d'un paquet avec cet outil, npx va chercher dans la variable "PATH" de l'ordinateur puis dans les fichiers binaires des modules du projet pour lancer la commande. S'il ne l'a pas trouvé, l'outil est même capable d'aller sur internet chercher la commande et de l'exécuter ensuite. Le paquet est exécuté dans le répertoire en cours.

IV. L'outil create react app

Create React App va vous permettre de **générer un squelette de code** pour votre application React.

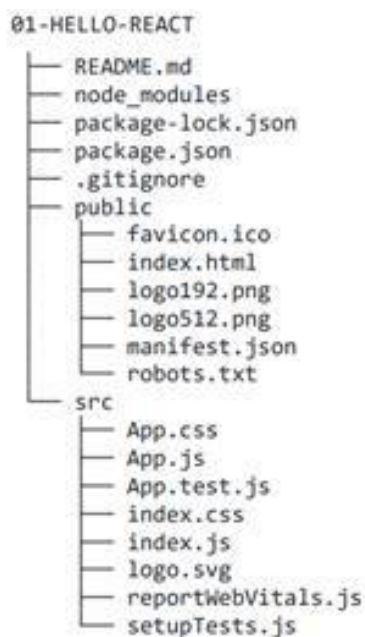
V. Première application

Voici la commande pour créer une nouvelle application react en utilisant le package **create-react-app** : **npx create-react-app nom-app**

La commande pour lancer votre application est npm start.

VI. Structure d'un projet React

Voici la structure du projet créé par l'outil create-react-app :



Vous trouverez trois dossiers 📁:

- **node_modules** : C'est là que sont installées toutes les **dépendances** de notre code.
- **Public** : Vous y trouverez le fichier [index.html](#) qui est notre point départ avec la div qui a pour class root qui permet de signifier a React ou est ce qu'il doit venir s'ancrer.
- **src** : vous venez de rentrer dans le cœur de l'action. **L'essentiel des fichiers que vous créerez et modifierez seront là**. Ce dossier contient aussi le fichier index.js qui est le fichier qui charge notre premier composant, le composant App qui est ensuite greffer a la div avec la class root qui se trouve dans le fichier index.html au niveau du dossier public.
- **Un fichier package.json** :
 - Liste les paquets dont dépend votre projet
 - Spécifie les versions d'un paquet que votre projet peut utiliser.
 - Rendre possible la reconstruction du projet, et donc plus facile à partager avec d'autres développeurs.
- **Le fichier .gitignore** a pour rôle de signifier a git les fichiers qui ne doivent pas être pusher sur le repo distant notamment le dossier node module.

Explication du code

Quand vous créez un projet React avec create-react-app, il y a deux fichiers principal, le fichier public/index.html et le fichier src/index.js

Le fichier public/index.html : Ce fichier ne contient pas grand-chose, c'est là qu'est défini la structure de base d'une page HTML, plus important il y est un élément dic défini avec un id root.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
    <title>React App</title>
```

```
</head>
<body>
  <noscript>You need to enable JavaScript to run this app.</noscript>
  <div id="root"></div>
</body>
</html>
```

C'est en effet dans cette div que notre application sera affichée.

Le fichier index.js : Ce fichier va permettre de récupérer nos composants et les afficher dans la div#root de public/index.html.

```
1  import React from 'react';
2  import ReactDOM from 'react-dom/client';
3  import './index.css';
4  import App from './App';
5  import reportWebVitals from './reportWebVitals';
   const root =
   ReactDOM.createRoot(document.getElementById('root'));
   root.render(
6    <React.StrictMode>
      <App />
    </React.StrictMode>
   );
7  reportWebVitals();
```

Les lignes 1 et 2 permettent l'importation des modules react et ReactDOM.

La ligne 3 permet l'importation du fichier de style index.css.

La ligne 4 : importe le fichier src/App.js qui définit une fonction nommée App.

La ligne 6 : affiche l'élément React au sein d'un nœud DOM root et renvoie une référence sur le composant.

Le fichier App.js

```
import './App.css';
import Content from './components/Content/Content';
import Footer from './components/Footer/Footer';
import Header from './components/Header/Header';
import Navigation from './components/Navigation/Navigation';
```

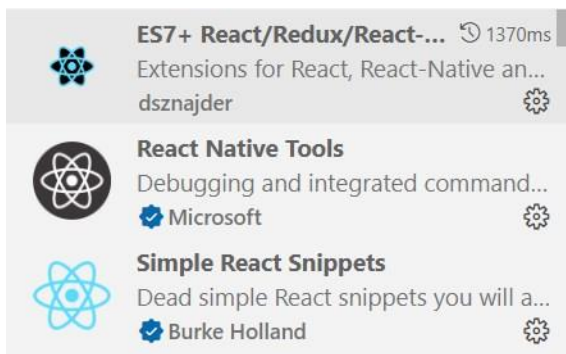
```
function App() {  
  return (  
    <div className="container">  
      <Header/>  
      <Navigation />  
      <Content />  
      <Footer />  
    </div>  
  );  
}  
  
export default App;
```

La fonction App renvoie une expression JSX. Cette expression définit ce que votre navigateur restitue finalement au DOM.

Tout en bas du fichier App.js, l'instruction export default App rend notre composant App disponible pour les autres modules

VII. Les extensions de VS code

Ils existent plusieurs extensions de react sous VS code, les plus utilisées sont :



VIII. Les extensions de chrome

L'extension de react sous chrome la plus utilisée est : **React Developer Tools**

React Developer Tools est une extension Chrome DevTools pour la bibliothèque JavaScript open-source React. Elle vous permet d'inspecter les hiérarchies des composants React dans les outils de développement de Chrome.

IX. Exercice d'application :

Créer un projet react

- 1- Créer un projet react avec l'outil create react app
- 2- Explorer la structure de votre projet
- 3- Lancer votre application.