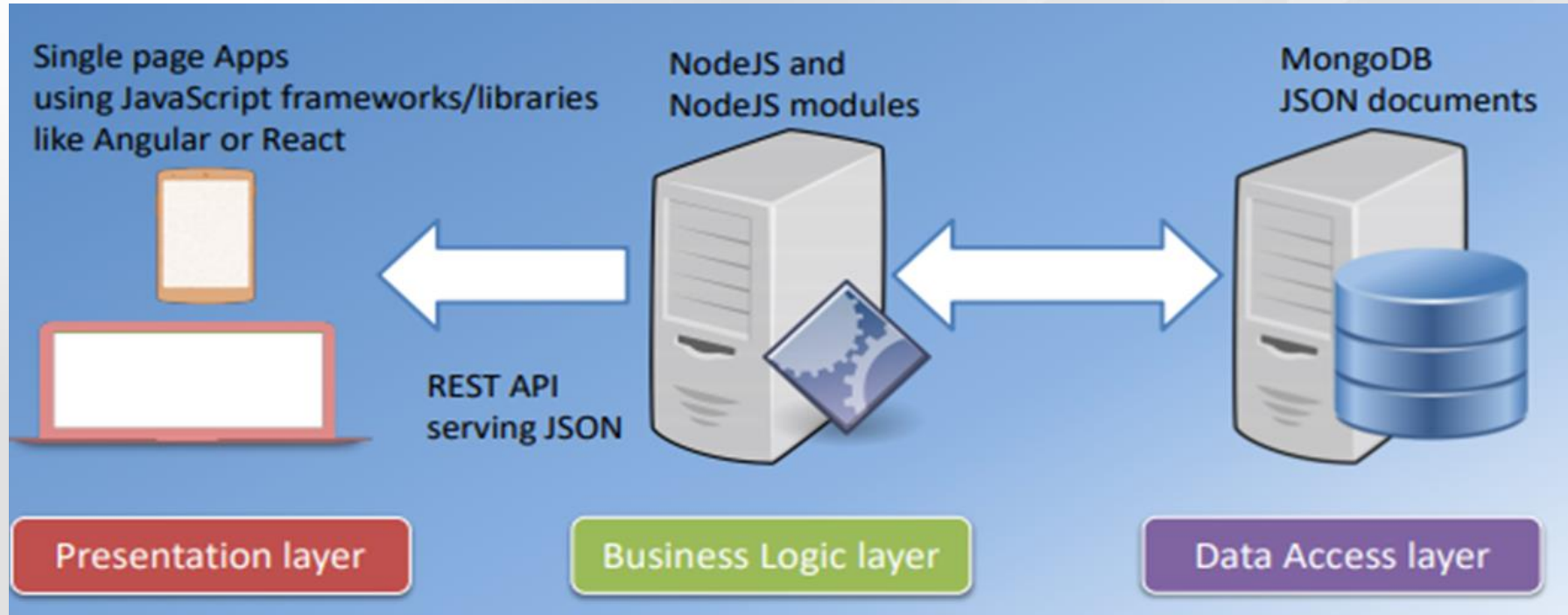


Chapitre 1 : Introduction et mise en place

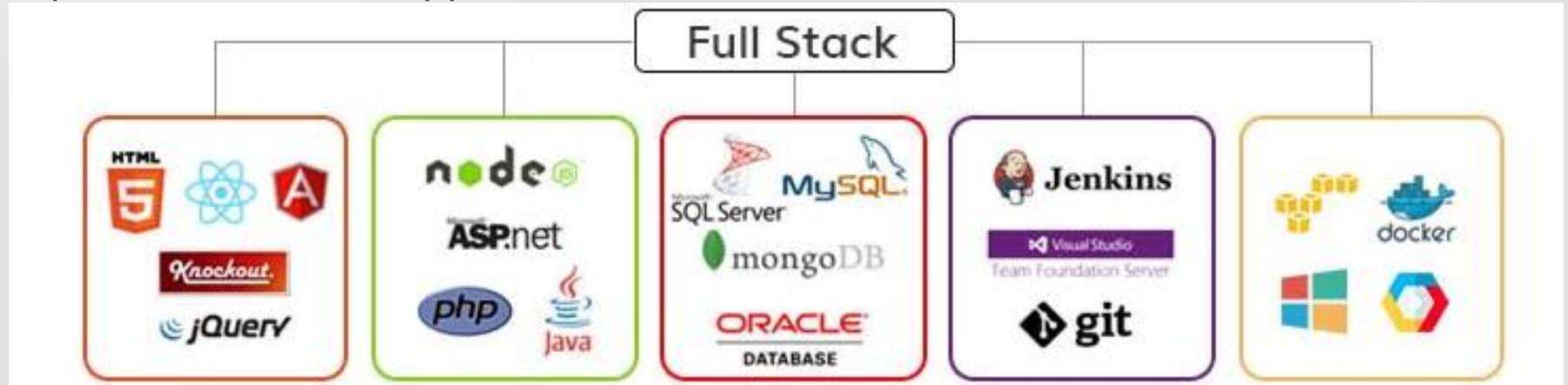


Architecture Full-stack



Architecture Full-stack

- ▶ Un développeur full-stack possède les connaissances complètes pour développer une application Web, une interface utilisateur ou des API sur un serveur.
- ▶ Cela signifie qu'ils sont bons pour le développement backend, le développement frontend, la gestion de bases de données, le développement d'applications mobiles et les tests.
- ▶ Ces polyvalents ont une connaissance de tout ce qui concerne l'ensemble du processus de développement Web.



Présentation de ReactJS

- ▶ ReactJS est une librairie JavaScript permettant de faciliter la création de la partie Frontend des applications Web de type Single Page Application (SPA).
- ▶ Cette librairie est basée sur l'approche Web Components qui considère une page web comme un ensemble de composants, c'est à dire de parties unitaires que l'on va pouvoir facilement exploiter, réutiliser et tester.
- ▶ ReactJS est un projet open-source, conçu par Facebook et publiée en open source depuis 2013, ReactJS permet de concevoir des applications web très performantes.
- ▶ Les applications React sont écrites en JSX (JavaScript Extension) qui est une extension de la syntaxe du langage JavaScript avec un moteur de Template permettant de générer et de manipuler le DOM d'une manière simple et efficace.

Intérêts

- Approche « Composant », différent des autres (Angular 1, Angular 2, Ember, etc.).
- Au lieu de modifier le DOM, on écrit du HTML dans les composants JavaScript en interne.
- Code plus compréhensible.
- Intègre parfaitement les nouveautés ECMAScript 6 (alias JavaScript 2015).
- Existence de React Native → permettant de créer des applications mobiles à la manière de Cordova ou Ionic pour Angular.
- De plus, React a été créé en se concentrant sur les performances, utilisant un DOM virtuel pour rafraîchir uniquement les composants qui ont été modifiés et non toute la page (il reste cependant plus lent que du JavaScript pur)

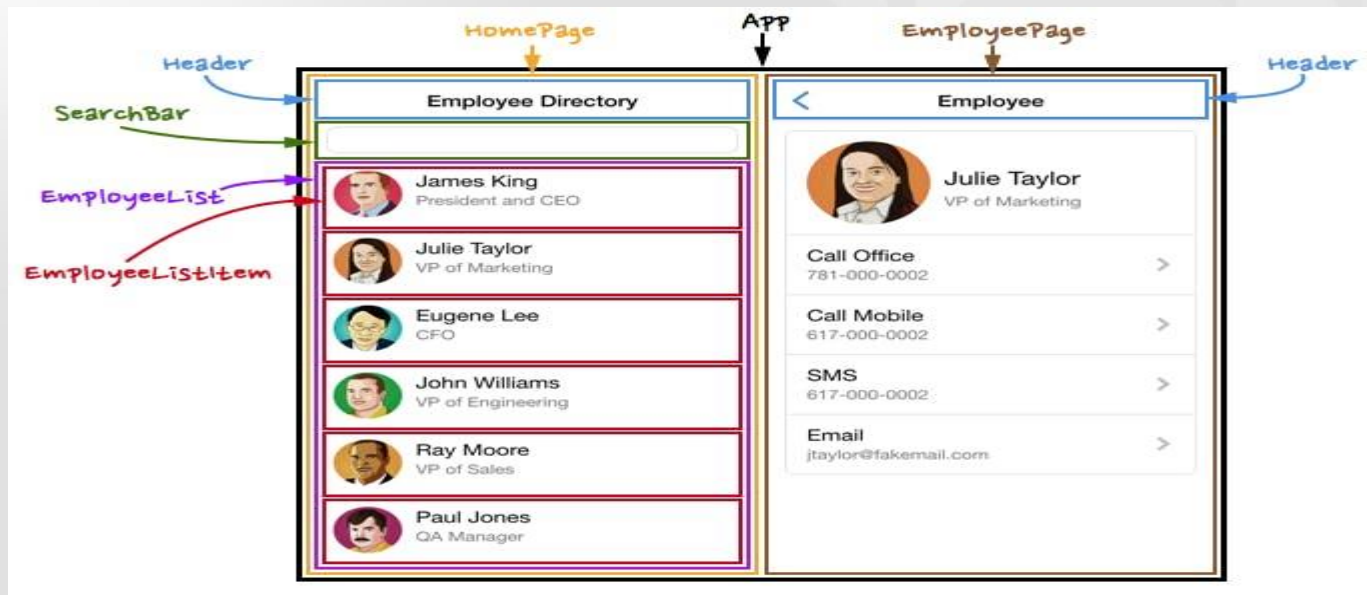
Composants

Découper l'interface utilisateur en éléments indépendants et réutilisables, vous permettant ainsi de considérer chaque élément de manière isolée.



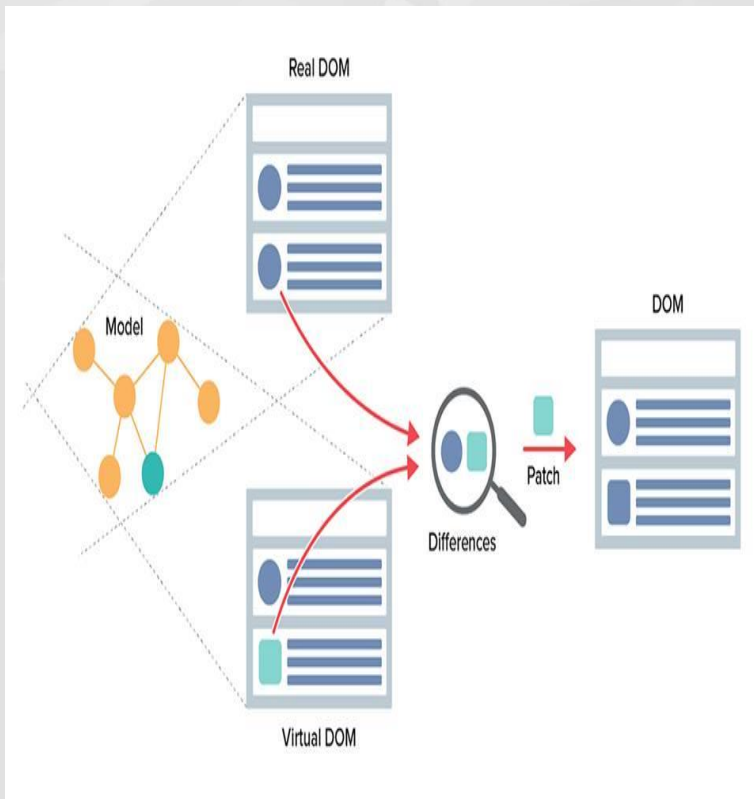
Composants réutilisables

- React permet aux développeur de créer des Component (Composants) correspondant aux parties de l'interface.
- Les composants peuvent être réutilisés ou combinés avec autres Components afin de créer une interface complète.



Virtual Dom

- Dans Virtual DOM, seule la différence entre les balises HTML précédentes et actuelles est observée et la différence est mise à jour.
- Chaque partie de l'interface utilisateur dans React est un composant et chaque composant a un état.
- Lorsqu'un état du composant est modifié, le DOM virtuel est mis à jour par React.
- Après avoir mis à jour le DOM virtuel, React compare la version actuelle du DOM virtuel avec la version précédente et recherche les objets qui ont été modifiés.
- Ces objets modifiés sont ensuite mis à jour dans le vrai DOM.
- Cela réduit le temps d'exécution et offre des performances élevées.



Avantages

- ▶ React ouvre les portes des applications universelles qui se basent sur les pages SPA (Single Page Application – s'exécute entièrement dans le navigateur).
 - Navigation plus fluide
 - Serveur moins sollicité (données récupérées par des appels API)
 - Interactions plus riches
- ▶ React Native : il offre la possibilité d'intégrer facilement une application Web développée avec ReactJs dans un environnement natif IOS ou Android.

Inconvénients

- ▶ L'écosystème est faible par rapport aux plus anciens tels qu'Angular, etc.
- ▶ Il n'y a aucune garantie que le code écrit à l'heure actuelle fonctionnera avec les futures versions de React.
- ▶ L'approche de l'interface utilisateur est complètement différente.

Create-React-App

- Create-React-App est un outil pour faciliter le développement d'applications web fondées sur React.
- L'idée est de générer automatiquement un squelette applicatif et de masquer la complexité potentielle de configuration des briques techniques associées.
- Il est par ailleurs facile de mettre à jour automatiquement l'outil lui-même, et gérer soi-même chaque aspect technique.

Conditions d'installation

- Nous aurons besoin de **Node** ≥ 8.10 et de **npm** ≥ 5.6 sur notre machine pour créer une application React.
- npm est le gestionnaire de modules de Node, qui facilite l'utilisation, les mises à jour et le partage de code JavaScript.
- npm est à la fois un outil en ligne de commande (la commande `npm`) et un référentiel en ligne de modules, interrogé par cet outil.
- Node.js est un environnement d'exécution JavaScript installable partout, qui permet d'écrire n'importe quel type de programme en JavaScript : outil en ligne de commande, serveur, application desktop, internet des objets (IoT), etc.
- Pour installer Node.js sous Windows, il suffit de télécharger l'installeur qui est proposé sur le site de Node.js : <https://nodejs.org/en/>

Download for Windows (x64)

| | |
|--|--|
| 12.18.3 LTS Recommended For Most Users | 14.9.0 Current Latest Features |
| Other Downloads Changelog API Docs | Other Downloads Changelog API Docs |

Or have a look at the [Long Term Support \(LTS\) schedule](#).

Installation 1/5

➤ Étape 1 - installez create-react-app

- Parcourez le bureau et installez l'application Create React à l'aide de l'invite de commande.
- On installe d'abord l'application create-react-app en exécutant cette commande dans le terminal :

```
C:\AppReact>npm install -g create-react-app
```

➤ Étape 2 – Créez une application avec create-react-app

- Exécutez cette commande pour créer une application React nommée myfirstapp.

Remarque :

Le nom de l'application doit être en minuscule

```
C:\TPReact>npx create-react-app myfirstapp
```

- Cela va créer un dossier nommé myfirstapp et y installer tous les fichiers requis.

Installation 2/5

- L'application create-react configurera tout ce dont vous avez besoin pour exécuter une application React.

```
Success! Created myfirstapp at C:\AppReact\myfirstapp
Inside that directory, you can run several commands:
```

```
npm start
```

```
Starts the development server.
```

```
npm run build
```

```
Bundles the app into static files for production.
```

```
npm test
```

```
Starts the test runner.
```

```
npm run eject
```

```
Removes this tool and copies build dependencies, configuration files
and scripts into the app directory. If you do this, you can't go back!
```

```
We suggest that you begin by typing:
```

```
cd myfirstapp
```

```
npm start
```

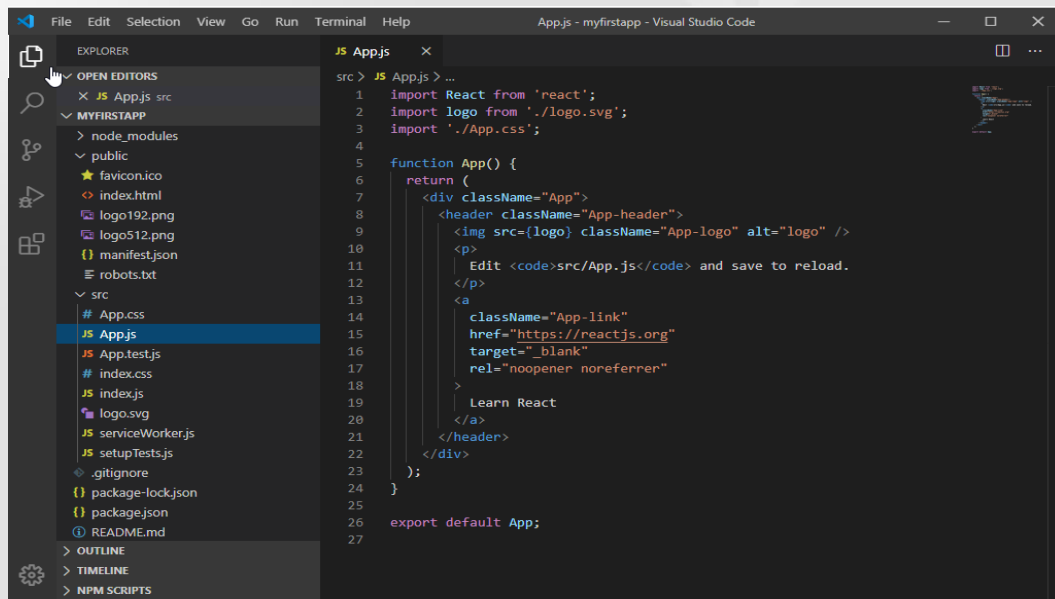
```
Happy hacking!
```


Installation 3/5

▶ Étape 3 - Ouvrez l'application React dans Visual Code

C:\AppRect> cd myfirstapp

C:\AppRect\myfirstapp>code .



Installation 4/5

► Étape 4 - Exécutez le projet

A l'aide de la commande : `npm start`

`C:\AppRect\myfirstapp>npm start`

```
Try the new cross-platform PowerShell https://aka.ms/pscore6  
Compiled successfully!
```

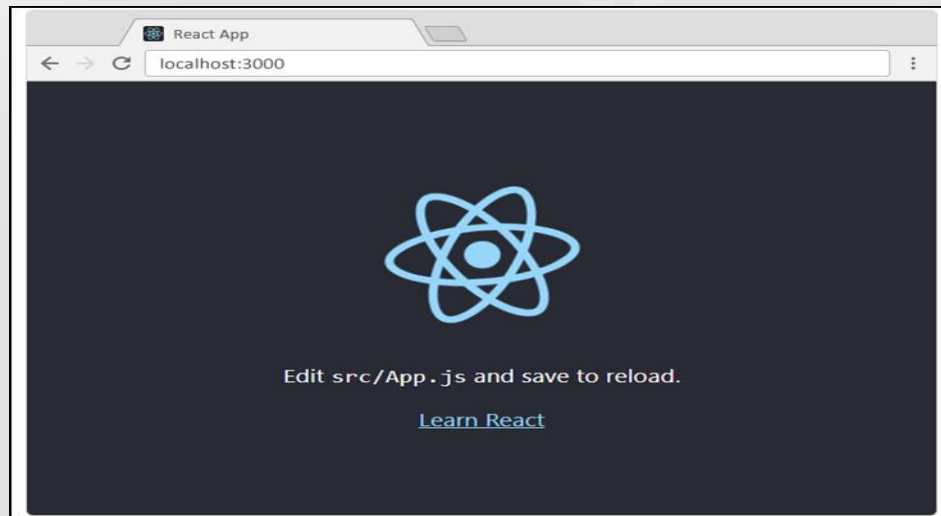
```
You can now view myfirstapp in the browser.
```

```
Local:          http://localhost:3000  
On Your Network: http://192.168.1.8:3000
```

```
Note that the development build is not optimized.  
To create a production build, use npm run build.
```

Installation 5/5

- ▶ Étape 5 - Visualisez le résultat dans le navigateur à l'adresse localhost:3000



Test d'un code

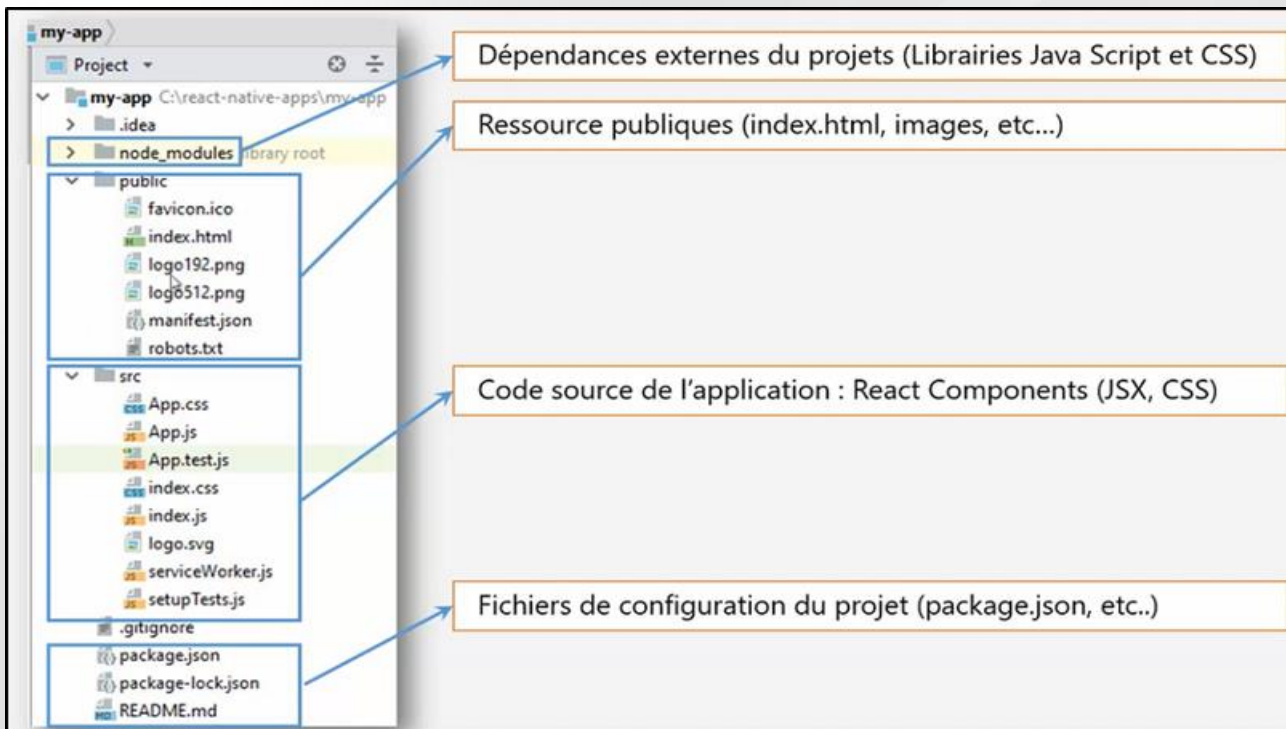
- Supprimez tous les fichiers source qui ne sont pas nécessaires :
 - Parcourez le dossier src dans le dossier my-app généré,
 - Supprimez les fichiers logo.svg, App.css et App.test.js.
- Dans le fichier src/App.js, remplacez le code existant par le code suivant :

```
import React, { Component } from 'react';  
class App extends Component {  
  render() {  
    return (  
      <div className="App">  
        <h1>Hello World!</h1>  
      </div>  
    );  
  }  
}  
export default App;
```

- Puis visualisez le résultat dans le navigateur :

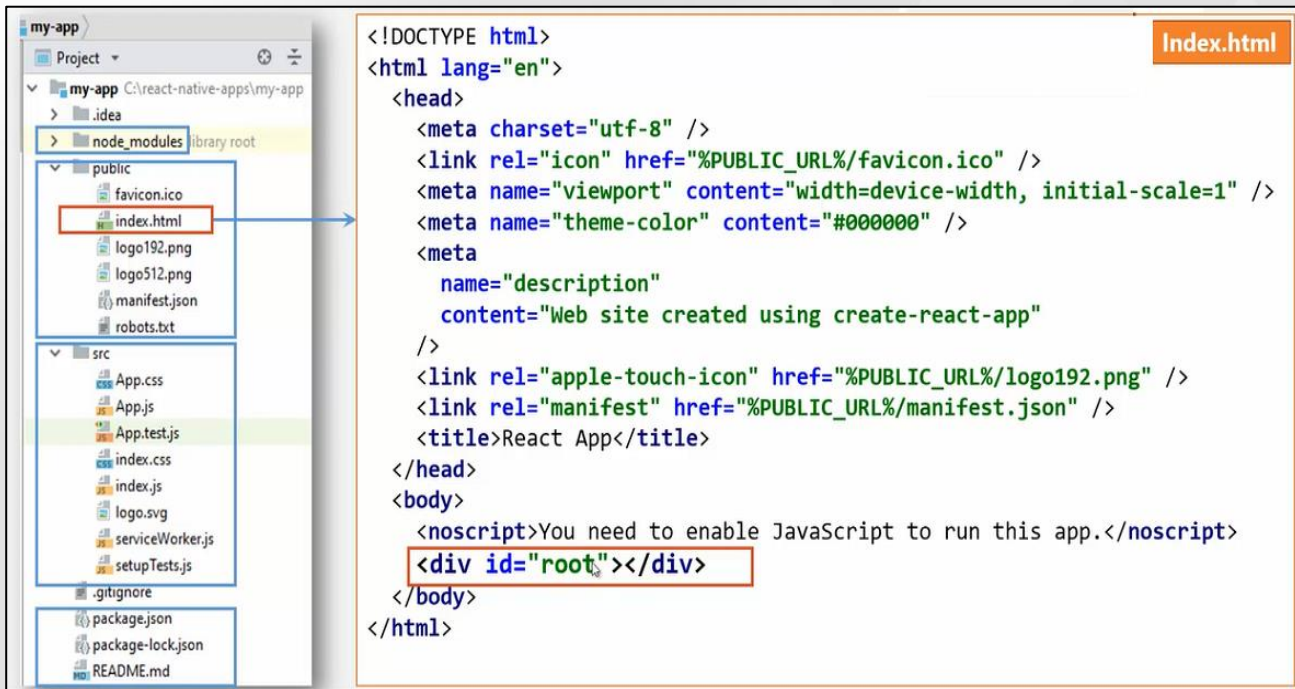


Structure d'un projet React



index.html

Si on ouvre le fichier public/index.html, on y trouve le contenu suivant :

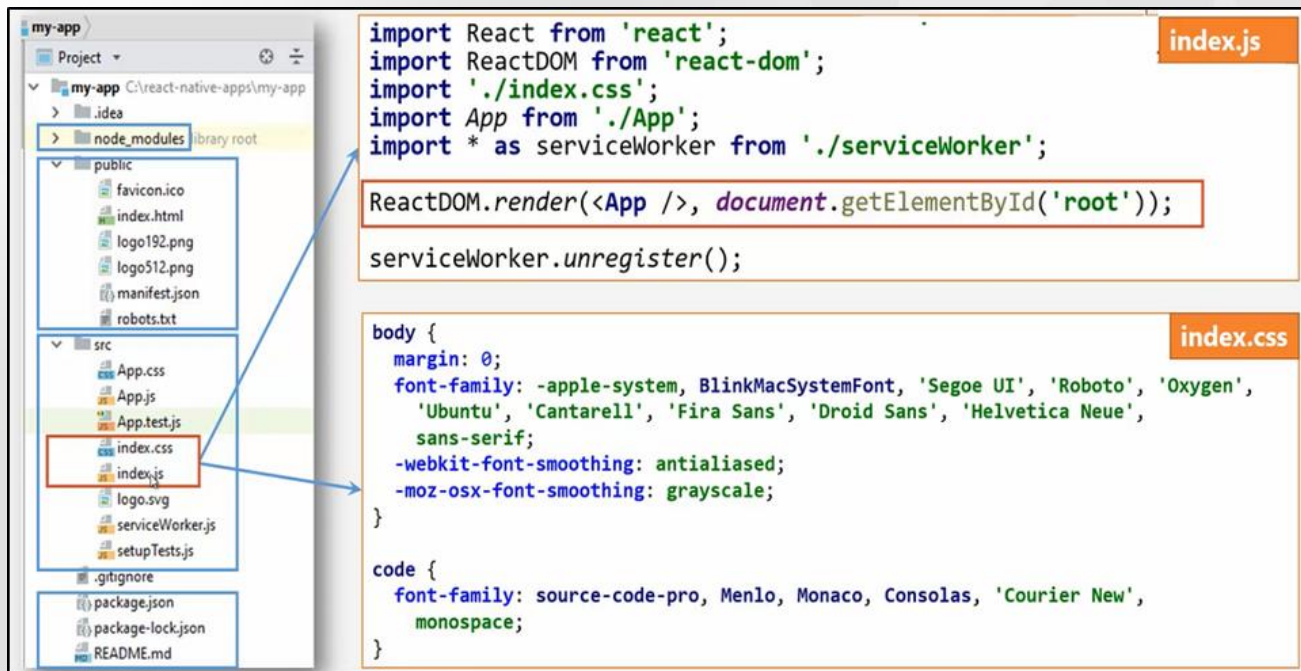


The image shows a screenshot of an IDE with a file explorer on the left and a code editor on the right. The file explorer shows the project structure, with the 'public' directory expanded. The 'index.html' file is highlighted with a red box, and a blue arrow points from it to the code editor. The code editor displays the content of 'index.html', which is a standard HTML document structure for a web application. The code is as follows:

```
Index.html
<!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>
```


index.js / index.css

Si on accède aux fichiers src/index.js et src/index.css, on y trouve le contenu suivant :



The image shows a code editor interface with a file explorer on the left and two code snippets on the right. The file explorer shows the project structure for 'my-app', with the 'src' directory expanded. The 'src' directory contains files like 'App.css', 'App.js', 'App.test.js', 'index.css', 'index.js', 'logo.svg', 'serviceWorker.js', and 'setupTests.js'. The 'index.js' file is highlighted with a red box, and a blue arrow points from it to the first code snippet. The 'index.css' file is also highlighted with a red box, and a blue arrow points from it to the second code snippet.

index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
import * as serviceWorker from './serviceWorker';

ReactDOM.render(<App />, document.getElementById('root'));

serviceWorker.unregister();
```

index.css

```
body {
  margin: 0;
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',
    'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
    sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}

code {
  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',
    monospace;
}
```

App.js

src/App.js est un exemple de composant React appelé «App» que nous obtenons par défaut lors de la création d'une nouvelle application :

