

Outils : create-react-app

Nous allons utiliser maintenant create-react-app qui nous permettra d'avoir un squelette d'application et un environnement de développement.

Ce module JS de développement utilise babel et webpack. Vous n'avez pas à priori à vous soucier des configurations de ces deux derniers packages.

Voici la documentation officielle de create-react-app

Pré-requis et présentation

Node et npm doivent être installés sur votre machine. Vérifiez que les versions de ces deux modules sont tels que : node \geq 8.10 et npm \geq 5.6

Puis installez de manière globale create-react-app :

```
npm install create-react-app --global
```

Si vous avez à mettre à jour votre create-react-app tapez la ligne de commande suivante :

```
npm install react-scripts@latest
```

Cet outil vous fournira les commandes principales ci-dessous :

```
# démarre le serveur de production
```

```
npm start
```

```
# build les sources dossier build
```

```
npm run build
```

```
# build en supprimant toutes les dépendances inutiles
```

```
# pas de retour en arrière possible
```

```
npm run eject
```

Installation d'un projet

Création du projet.

Tapez les lignes de code ci-dessous :

```
npm create-react-app denomination
```

```
cd denomination
```

```
# démarrer le serveur
```

```
npm start
```

npx est un exécuteur contenu dans npm depuis la version npm ≥ 5.2

Voici la structure d'un projet create-react-app :

```
denomination
|-- README.md
|-- node_modules    <-- dépendances
|-- package.json
|-- .gitignore
|-- public
|   |-- favicon.ico
|   |-- index.html    <-- Point d'entrée
|   |-- logo192.png
|   |-- logo512.png
|   |-- manifest.json
|   |-- robots.txt
|-- src
    |-- App.css
    |-- App.js        <-- composant racine
    |-- App.test.js
    |-- index.css
    |-- index.js      <-- bootstrap
    |-- logo.svg
    |-- serviceWorker.js
```

Projet dénomination

Pour faire ce projet vous allez utiliser un squelette d'application, tapez la ligne de commande suivante dans votre terminal :

```
npx create-react-app denomination
```

Puis placez-vous dans le dossier denomination et lancez votre serveur de développement :

```
cd denomination
```

```
npm start
```

Vous êtes prêt à travailler sur le projet. Lisez la suite.

Vous allez créer une petite application qui permet de saisir un montant donné et de fournir la monnaie que l'on peut rendre par rapport à ce montant.

Voici les dénominations, c'est-à-dire les tokens que l'on pourra rendre en fonction du montant :

```
const denominations = [1, 5, 10, 20, 50, 100, 200] ;
```

Une fois le montant proposé, par exemple 233, affichez les tokens correspondants :

```
token 200 : 1
token 20  : 1
token 10  : 1
token 1   : 3
```

Un montant ne pourra être inférieur à 1.

Wireframe saisir un montant

```
saisir le montant [ 233 ]
```

```
[ generate token(s) ]
```

Wireframe token(s) rendu(s)

Vous avez demandé la monnaie sur 233 :

```
token 200 : 1
```

```
token 20  : 1
```

```
token 10  : 1
```

```
tokens 1  : 3
```