

Développer avec React

Créer un nouveau projet

Développer avec React

Aperçu des différentes technologies

Il existe plusieurs solutions qui permettent de créer un projet avec React

- Web App Client
 - Vite
 - Create React App (CRA)
 - Parcel
- [Framework] Web App Server
 - Next.js
 - Remix
 - Gasby
- Intégration dans une page Web
 - CDN React
- [Framework] App Mobile
 - React Native
 - Expo
 - Ionic
- [Framework] App Desktop
 - Electron

Choix pour créer le projet

Avant de créer le projet, il faudra choisir la solution la plus adaptée à nos besoins :

- Application Web (Client ou Serveur)
- Application Mobile
- Application Desktop

Chacune de ces solutions ont bien entendu leurs avantages et leurs inconvénients.

Dans le cadre de l'apprentissage de React, nous allons créer un projet avec « Vite ».

Projet React avec Vite

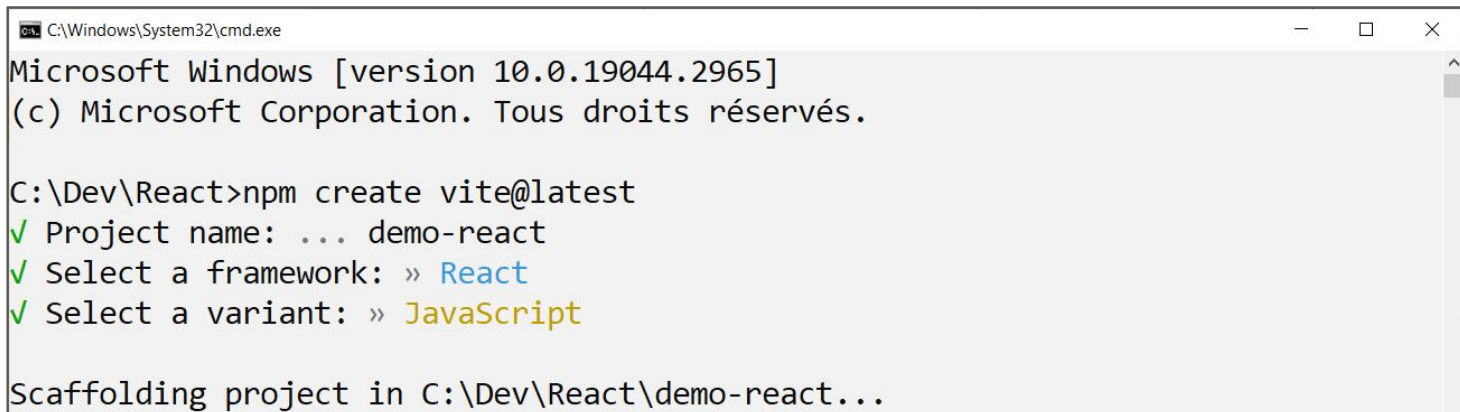
Développer avec React

Création de la base du projet

Ouvrir le terminal dans un répertoire.

Générer le projet en utilisant « Vite » avec le template « React + JavaScript »

⚠ Le nom du package doit être en minuscule et il ne doit pas contenir d'espace !



```
C:\Windows\System32\cmd.exe
Microsoft Windows [version 10.0.19044.2965]
(c) Microsoft Corporation. Tous droits réservés.

C:\Dev\React>npm create vite@latest
✓ Project name: ... demo-react
✓ Select a framework: » React
✓ Select a variant: » JavaScript

Scaffolding project in C:\Dev\React\demo-react...
```

Remarque : Remplacer le nom du projet par un point, pour le créer dans le répertoire actuel.

Création de la base du projet

Une fois la génération de code terminée, nous obtenons cette structure.

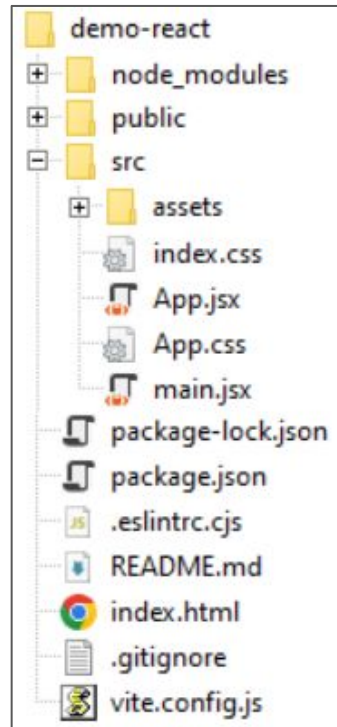
Pour tester le projet, nous allons démarrer le serveur de développement.

Pour cela, utiliser les commandes :

" cd <my-project> "

" npm install "

" npm run dev "



Explication des éléments générés

Quelques explications sur les éléments créés :

- Le dossier « public »
Ressources static (par exemple : une icône).
- Le dossier « src »
Code source de l'application React.
- Les fichiers « app.jsx »
Composant « **App** » de l'application
- Les fichiers « main.jsx »
Utilisation du **ReactDOM** pour injecter le composant « App » dans le code HTML.

Structure de fichier

React n'impose pas de structure pour vous ordonnez vos fichiers.

Il est conseillé d'en mettre une en place pour faciliter le développement et la maintenance des projets.

Des solutions populaires existent pour ordonner les fichiers d'un projet.

Par exemple :

- Par types d'éléments (components, store, api, page, ...)
- Par fonctionnalités

Structure de fichier - Choix pour le support

La structure utilisée dans le support est « Par types d'éléments », sous la forme :

- Un dossier par éléments.
- Les éléments seront séparé en catégories :
 - **Components** Composant "*simple*" et réutilisable.
Exemple → Une barre de recherche.
 - **Containers** Composant reprenant les différentes features.
Exemple → Le menu de navigation.

D'autres catégories de fichiers seront ajoutés par la suite (*page, store, api, ...*).

Le but des composants

Développer avec React

Les Composants React

Les composants React vont nous permettre de créer nos interfaces en blocs de code indépendant et réutilisable. Ceux-ci ont un paramètre d'entrée et un rendu.

Il existe deux familles de composant React :

- Les fonctions
 - Basique : limité à la génération de rendu via le JSX
 - Avancé : utilisation du mécanisme des Hooks
- Les classes héritant de « `React.Component` »

⚠ Lors du développement, nous allons créer un fichier par composant React. Pour cela, nous utiliserons les instructions d'import et d'export du JavaScript.

Découpe d'un rendu sous forme de composant

Pour concevoir une application Web en React.

Il est nécessaire de découper le rendu sous forme de différents composants.

Exemple de découpe :

- 1) **FilterableProductTable** : Le composant principal.
- 2) **SearchBar** : Interaction avec la saisie de l'utilisateur.
- 3) **ProductTable** : Affiche et filtre les données.
- 4) **ProductCategory** : Affiche le titre d'une catégorie.
- 5) **ProductRow** : Affiche de détail d'un produit.

