

Présentation React et ses fondamentaux

Découvrez React, la bibliothèque JavaScript créée par Facebook pour le développement d'interfaces utilisateur modernes et performantes. Nous allons explorer ensemble ses fondamentaux.



Nicolas Dobrowolski

Qu'est-ce que React ?

Définition de React

React est une **bibliothèque JavaScript** open-source qui permet de créer des interfaces utilisateur dynamiques et réutilisables.

Pourquoi utiliser React ?

Grâce à sa flexibilité et sa **facilité** d'utilisation, une grande **communauté**, gestion d'état efficace

Comprendre la philosophie de React

React est basé sur single-page application (**SPA**) et encourage la création de **composants** similaire a des briquets lego pour faciliter la **maintenance** et **l'évolutivité** du code.

Installation de React



Configuration de l'environnement de développement

Avant de créer un projet React, il est important de mettre en place un environnement de développement avec **Node.js** et le gestionnaire de paquets npm.



Création d'un projet React+ Vite

Un outil de développement qui **accélère** la création d'applications React grâce à un rechargement **instantané**, démarrage instantané, et une performance **optimale** en production.

Composants React

1

Composants et rendu

Les composants sont les fondements de l'architecture React. Ils peuvent être créés facilement avec une syntaxe similaire à HTML et sont ensuite rendus dans le DOM.

2

Props (Propriétés)

Les props permettent de passer des données entre les composants et de personnaliser leur apparence et leur comportement en temps réel.

3

State (État)

L'état est un autre concept fondamental dans React. Il permet de stocker les données qui doivent être modifiées et d'actualiser l'interface en temps réel en fonction de ces changements.

Création de composants

```
src > App.jsx > default
nico.dobrowolski, il y a 17 heures | 1 author (nico.dobrowolski)
1 import { BrowserRouter as Router, Routes, Route } from "react-router-dom";
2 import ViewLogin from "../views/ViewLogin/ViewLogin";
3 import Header from "../components/common/header/Header";
4
5 Codeium: Refactor | Explain | Generate JSDoc
6 const App = () => {
7   return (
8     <Router>
9       <Header />
10      <Routes>
11        /* Make private route */
12        <Route path="/login" element={<ViewLogin />} />
13        <Route path="*" element={<h1>Not found</h1>} />
14      </Routes>
15    </Router>
16  );
17 };
18
19 export default App; nico.dobrowolski, hier • Add and config Mui
21
```

Création de composants

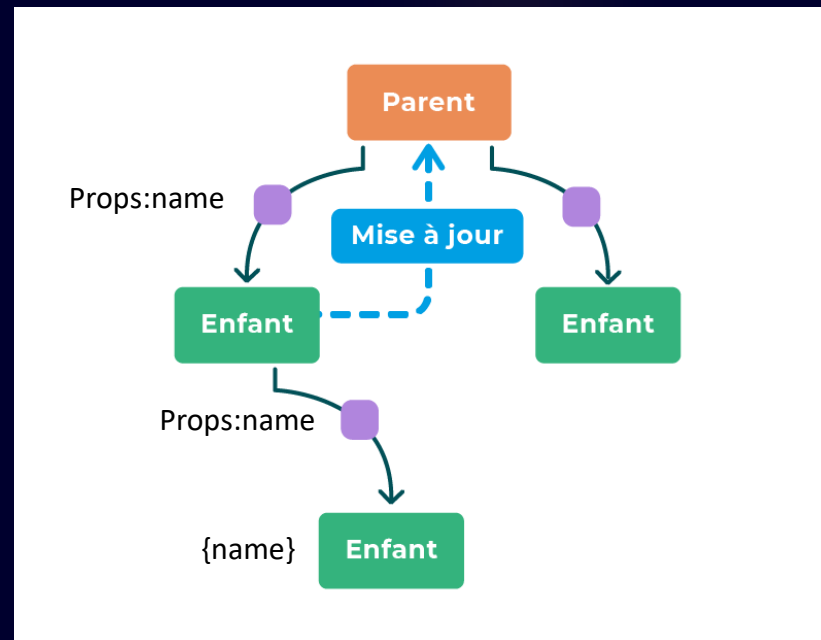
La création de composants en React consiste à définir des éléments d'interface utilisateur qui encapsulent la logique et la présentation liées à une partie spécifique de votre application.



Réutilisabilité des composants

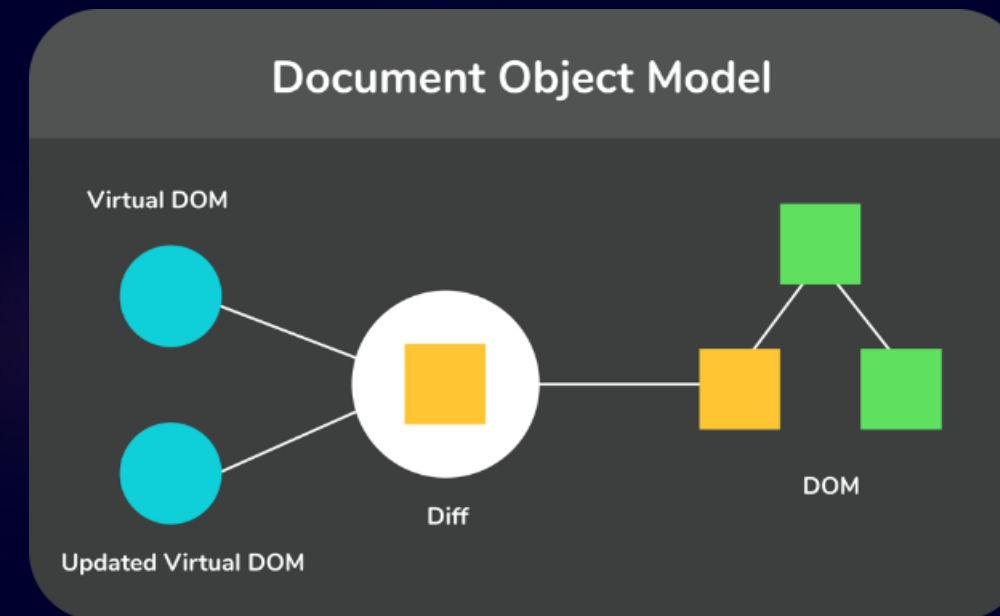
Grâce à la modularité des composants, ces derniers peuvent être réutilisés dans plusieurs parties de l'application pour améliorer la lisibilité et la structure du code.

Passage de données entre les composants



Passage de données entre composants

Le passage de données entre les composants en React permet de transmettre des informations d'un composant parent à un composant enfant via les propriétés (props) pour personnaliser et partager des données entre eux.



Avantages du DOM virtuel

Le DOM virtuel permet une mise à jour efficace et rapide de l'interface utilisateur en ne mettant à jour que les éléments qui ont changé, réduisant ainsi les coûts en performances et en maintenance.

Gestion de l'état dans un composant

Utilisation de useState

useState est un Hook de React qui permet de gérer l'état local dans un composant et de déclencher automatiquement le rendu de l'interface en fonction des modifications apportées à l'état.

1

Comprendre l'état

L'état est un objet Javascript mutable qui stocke les données nécessaires à l'interface utilisateur et peut être modifié à tout moment.

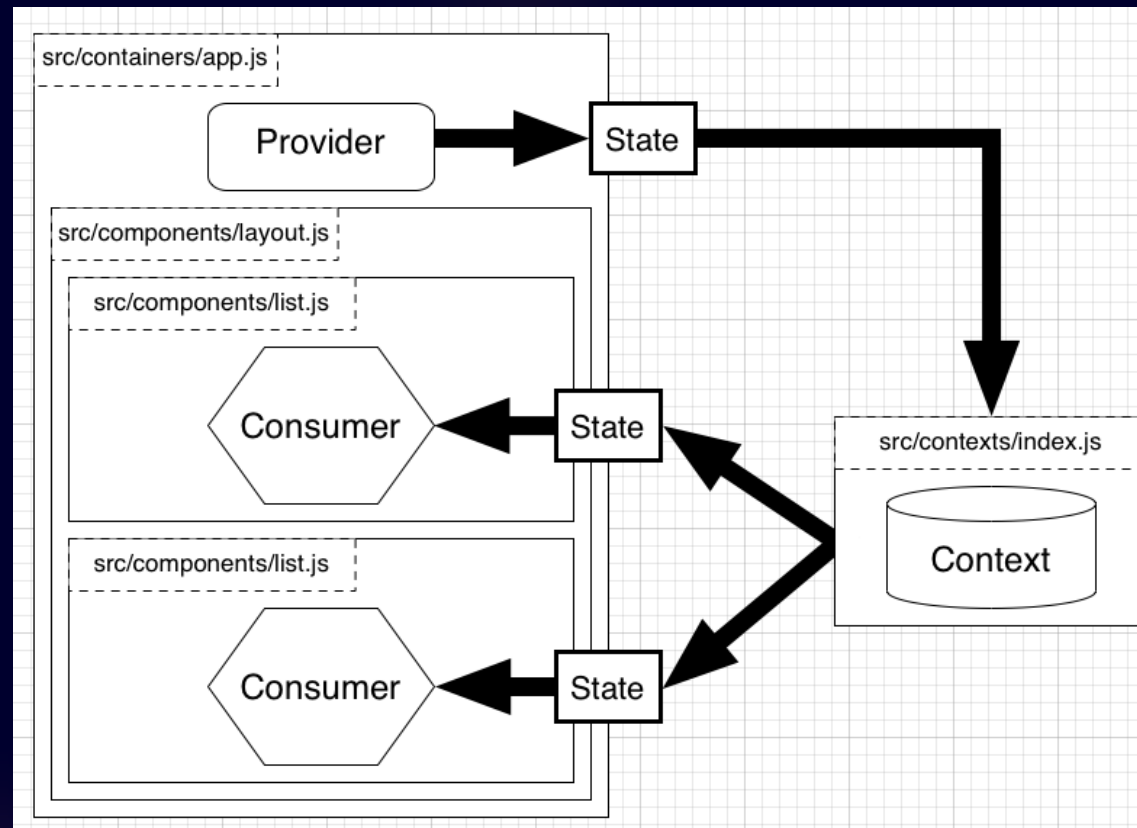
2

3

Utilisation de useEffect

Permet d'effectuer des actions dans un composant en réponse à des changements d'état ou de données spécifiques.

Passage de données dans un contexte



Contexte

Le Contexte offre un moyen de faire passer des données à travers l'arborescence du composant sans avoir à passer manuellement les props à chaque niveau.