

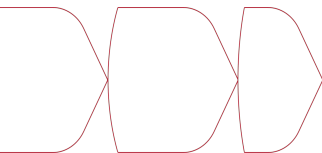
Introduction à React

Comprendre les bases de la bibliothèque JavaScript pour les interfaces utilisateur

Slimani Mohamed Amine

EHTP

February 14, 2025



Sommaire

Qu'est-ce que React ?

Pourquoi utiliser React ?

Concepts de base de React

Exemple de composant React

Gestion de l'état avec useState

Bonnes pratiques

Outils pour travailler avec React

Exemple d'application React

Défis de React

Pourquoi c'est important ?

Qu'est-ce que React ?

- ▶ **Définition** : React est une bibliothèque JavaScript open-source développée par Facebook pour construire des interfaces utilisateur.
- ▶ **Objectif** : Permettre de créer des applications web interactives et performantes.
- ▶ **Avantages** : Composants réutilisables, DOM virtuel, et grande communauté.

Pourquoi utiliser React ?

- ▶ **Composants réutilisables** : Permet de créer des composants modulaires et réutilisables.
- ▶ **Performance** : Utilisation du DOM virtuel pour des mises à jour efficaces.
- ▶ **Écosystème riche** : De nombreuses bibliothèques et outils disponibles.

Concepts de base de React

- ▶ **Composants** : Blocs de construction de base d'une application React.
- ▶ **JSX** : Syntaxe qui permet d'écrire du HTML dans JavaScript.
- ▶ **État (State)** : Données internes d'un composant qui peuvent changer.
- ▶ **Props** : Données passées d'un composant parent à un composant enfant.

Exemple de composant React

Composant fonctionnel en React

```
import React from 'react';

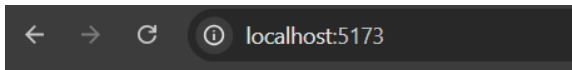
Comment Code
function Welcome(props) {
  return <h1>Bonjour, {props.name}</h1>;
}

export default Welcome;
```

```
import Welcome from './Welcome'

Comment Code
function App() {
  return (
    <>
      <Welcome name="Amine" />
    </>
  )
}

export default App
```



Bonjour, Amine!

Gestion de l'état avec useState

Utilisation de useState pour gérer l'état

```
import React, { useState } from 'react';

Comment Code
function Counter() {
  const [count, setCount] = useState(0);

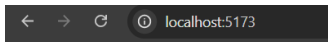
  return (
    <div>
      <p>Vous avez cliqué {count} fois</p>
      <button onClick={() => setCount(count + 1)}>
        Cliquez ici
      </button>
    </div>
  );
}

export default Counter;
```

```
import Counter from "../Counter"

Comment Code
function App() {
  return (
    <>
      <h1>Compteur</h1>
      <Counter />
    </>
  )
}

export default App
```



Compteur

Vous avez cliqué 7 fois

Cliquez ici

Effets secondaires avec useEffect

Utilisation de useEffect pour les effets secondaires

```
import React, { useState, useEffect } from 'react';
```

Comment Code

```
function Timer() {  
  const [seconds, setSeconds] = useState(0);  
  
  useEffect(() => {  
    const interval = setInterval(() => {  
      setSeconds(seconds => seconds + 1);  
    }, 1000);  
    return () => clearInterval(interval);  
  }, []);  
  
  return <div>Secondes : {seconds}</div>;  
}
```

export default Timer;

```
import Timer from './Timer';
```

Comment Code

```
function App() {  
  return (  
    <>  
      <h1>Timer</h1>  
      <Timer />  
    </>  
  )  
}  
  
export default App
```

← → ↺ Ⓞ localhost:5173

Timer

Secondes : 9

Bonnes pratiques

- ▶ **Composants purs** : Évitez les effets secondaires dans les composants.
- ▶ **État local** : Utilisez l'état local uniquement lorsque nécessaire.
- ▶ **Modularité** : Divisez l'application en petits composants réutilisables.

Outils pour travailler avec React

- ▶ **Create React App** : Outil pour créer rapidement des applications React.
- ▶ **React DevTools** : Extension de navigateur pour déboguer les applications React.
- ▶ **Redux** : Bibliothèque pour la gestion de l'état global.

Exemple d'application React

Application simple de liste de tâches

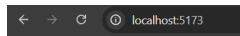
```
import React, { useState } from 'react';

function TodoApp() {
  const [tasks, setTasks] = useState([]);
  const [input, setInput] = useState('');

  const addTask = () => {
    setTasks([...tasks, input]);
    setInput('');
  };

  return (
    <div>
      <input value={input} onChange={(e) => setInput(e.target.value)} />
      <button onClick={addTask}>Ajouter</button>
      <ul>
        {tasks.map((task, index) => <li key={index}>{task}</li>)}
      </ul>
    </div>
  );
}

export default TodoApp;
```



Todo App

- lire un livre

Défis de React

- ▶ **Courbe d'apprentissage** : React peut être complexe pour les débutants.
- ▶ **Gestion de l'état** : La gestion de l'état peut devenir complexe dans les grandes applications.
- ▶ **Performances** : Optimisation nécessaire pour les applications très interactives.

Pourquoi c'est important ?

- ▶ React est largement utilisé dans l'industrie pour construire des interfaces utilisateur modernes.
- ▶ Il permet de créer des applications web performantes et maintenables.
- ▶ Comprendre React est essentiel pour les développeurs front-end.

Résumé

React est une bibliothèque puissante pour construire des interfaces utilisateur interactives. Explorez, apprenez, et construisez avec React !