

Créer un front en React avec les données d'une API

Étape 1 : Installer Node.js et npm

Pour commencer avec React, tu dois avoir Node.js et npm (le gestionnaire de paquets pour Node.js) installés sur ton ordinateur. Tu peux vérifier si tu les as déjà installés en exécutant les commandes suivantes dans ton terminal :

```
node -v  
npm -v
```

Si ces commandes renvoient des versions, Node.js et npm sont déjà installés. Sinon, télécharge-les et installe-les depuis [le site officiel de Node.js](https://nodejs.org/fr/).

Étape 2 : Créer un projet React avec Create React App

La manière la plus simple de commencer avec React est d'utiliser l'outil Create React App, qui va te générer une application React de base.

1. **Ouvre un terminal.**
2. **Navigue** vers le répertoire où tu veux créer ton projet.
3. **Exécute la commande suivante :**

Attention la commande suivante est dépréciée

```
npx create-react-app front-react
```

Cela va créer un dossier front-react avec tous les fichiers nécessaires pour commencer un projet React.

Attention à ne pas mettre de majuscules à votre Projet : React l'interdit.

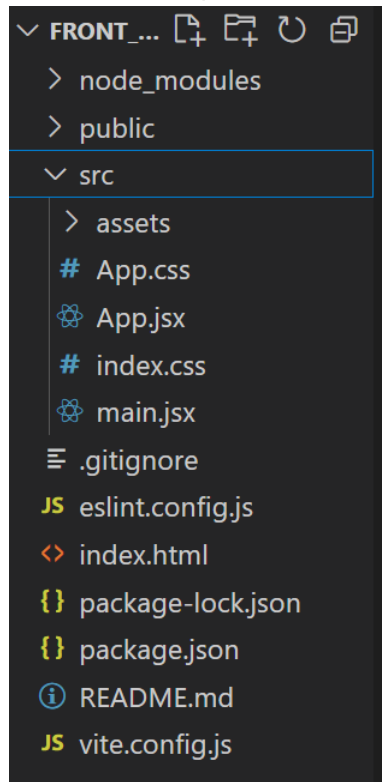
A la place, utiliser :

```
PS E:\Cours\DWMM\1.5 - Front End vers backend après mongo\Framework JS> npm create vite@latest  
Need to install the following packages:  
create-vite@5.5.2  
Ok to proceed? (y) y  
✓ Project name: ... front_react  
? Select a framework: » - Use arrow-keys. Return to submit.  
  > Vanilla  
    Vue  
    React  
    Preact  
    Lit  
✓ Select a framework: » React  
✓ Select a variant: » JavaScript  
  
Scaffolding project in E:\Cours\DWMM\1.5 - Front End vers backend après mongo\Framework JS\front_react...  
  
Done. Now run:  
  
  cd front_react  
  npm install  
  npm run dev
```

Installons bootstrap

```
npm install bootstrap
```

La création du projet peut être assez longue (ne paniquez pas !!!!). A la fin on a une arborescence qui ressemble à cela :



4. **Navigue dans le dossier de ton projet :**

```
cd front_react
```

5. **Lance le serveur de développement :**

```
npm run dev
```

Cela ouvrira une fenêtre de navigateur avec l'application React par défaut.



Vite + React

count is 0

Edit src/App.jsx and save to test HMR

[Click on the Vite and React logos to learn more](#)

React

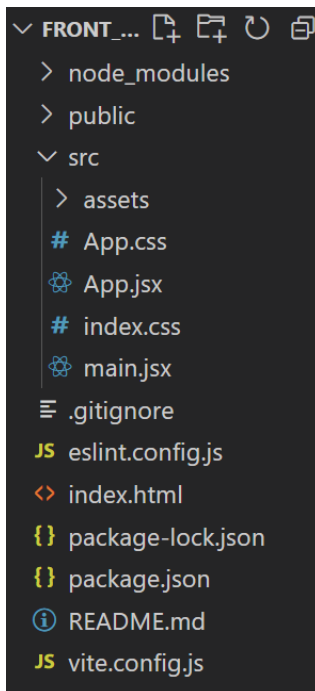
Réalisation : Guillaume DELACROIX Formateur
AFPA

26 août 2024

Afpa

Étape 3 : Comprendre la structure des fichiers

Voici les fichiers et dossiers les plus importants dans un projet React :



- **index.html** : Le point d'entrée principal de votre application.
- **src/main.jsx** : Le fichier où React est initialisé et lié au DOM.
- **src/App.jsx** : Le composant principal qui servira de conteneur pour d'autres composants.
- **src/components/** : Créez ce dossier pour vos composants réutilisables.
- **src/services/** : Créez ce dossier pour vos services API (comme la gestion des produits)

Dans un premier temps, importez Bootstrap dans votre projet. Modifiez **src/main.jsx** pour inclure les fichiers CSS de Bootstrap :

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';
import 'bootstrap/dist/css/bootstrap.min.css'; // Import Bootstrap
import './index.css';

ReactDOM.createRoot(document.getElementById('root')).render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
);
```

Étape 4 : Créer un Fichier `constants.js`

Créez un fichier `constants.js` pour centraliser l'URL de votre API.

Chemin : `src/constants.js`

```
export const API_ENDPOINT = 'http://localhost:3000';
```

Étape 5 : Créer les Services API

Créez un service pour gérer les produits et extraire les catégories à partir des produits existants.

Chemin : `src/services/productService.js`

```
import { API_ENDPOINT } from '../constants';

export const getProducts = async () => {
  const response = await fetch(`${API_ENDPOINT}/products`);
  return response.json();
};

// Extraire les catégories des produits existants
export const getCategoriesFromProducts = async () => {
  const products = await getProducts();
  const categories = products.map(product => product.category);

  // Éliminer les doublons de catégories en utilisant l'_id
  const uniqueCategories = Array.from(new Set(categories.map(cat => cat._id)))
    .map(id => categories.find(cat => cat._id === id));

  return uniqueCategories;
};

export const addProduct = async (product) => {
  const response = await fetch(`${API_ENDPOINT}/products/add`, {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    body: JSON.stringify(product),
  });
  return response.json();
};
```

Étape 6 : Créer le Composant `AddProduct`

Ce composant permet d'ajouter un produit en sélectionnant une catégorie existante ou en créant une nouvelle catégorie.

Chemin : `src/components/AddProduct.jsx`

```
import React, { useState, useEffect } from 'react';
import { getCategoriesFromProducts, addProduct } from
'../services/productService';

function AddProduct() {
  const [productName, setProductName] = useState('');
  const [productPrice, setProductPrice] = useState('');
  const [newCategoryName, setNewCategoryName] = useState('');
  const [selectedCategoryId, setSelectedCategoryId] = useState('');
  const [categories, setCategories] = useState([]);

  useEffect(() => {
    // Récupère toutes les catégories disponibles à partir des produits
    getCategoriesFromProducts().then(setCategories);
  }, []);

  const handleSubmit = async (e) => {
    e.preventDefault();

    let category = { _id: selectedCategoryId };

    if (newCategoryName) {
      // Génère un ID pour la nouvelle catégorie si un nom est fourni
      category = {
        _id: generateCategoryId(), // Génère un ID aléatoire
        name: newCategoryName,
      };
    } else {
      const selectedCategory = categories.find(cat => cat._id ===
selectedCategoryId);
      if (selectedCategory) {
        category.name = selectedCategory.name;
      }
    }

    const product = {
      name: productName,
      price: parseFloat(productPrice),
      category,
    };

    await addProduct(product);

    // Réinitialiser le formulaire
    setProductName('');
    setProductPrice('');
    setNewCategoryName('');
    setSelectedCategoryId('');
  };
}
```

React

Réalisation : Guillaume DELACROIX Formateur
AFPA

26 août 2024

The logo for Afpa (Association Française pour le Développement des Professions et de l'Apprentissage) features the word "Afpa" in a bold, black, sans-serif font. To the right of the text is a green circular graphic element.

```

return (
  <div className="container mt-5">
    <h2>Add Product</h2>
    <form onSubmit={handleSubmit}>
      <div className="mb-3">
        <label htmlFor="productName" className="form-label">Product Name</label>
        <input
          type="text"
          className="form-control"
          id="productName"
          value={productName}
          onChange={(e) => setProductName(e.target.value)}
        />
      </div>
      <div className="mb-3">
        <label htmlFor="productPrice" className="form-label">Price</label>
        <input
          type="number"
          className="form-control"
          id="productPrice"
          value={productPrice}
          onChange={(e) => setProductPrice(e.target.value)}
        />
      </div>
      <div className="mb-3">
        <label htmlFor="category" className="form-label">Category</label>
        <select
          className="form-select"
          id="category"
          value={selectedCategoryId}
          onChange={(e) => setSelectedCategoryId(e.target.value)}
          disabled={!newCategoryName}
        >
          <option value="">Select an existing category</option>
          {categories.map(category => (
            <option key={category._id} value={category._id}>
              {category.name}
            </option>
          ))}
        </select>
      </div>
      <div className="mb-3">
        <label htmlFor="newCategoryName" className="form-label">Or Create New
Category</label>
        <input
          type="text"
          className="form-control"
          id="newCategoryName"
          value={newCategoryName}
          onChange={(e) => {
            setNewCategoryName(e.target.value);
            setSelectedCategoryId(''); // Réinitialiser la catégorie
sélectionnée si une nouvelle catégorie est entrée
          }}
          placeholder="Enter new category name"
        />
      </div>
    </form>
  </div>
);

```

React

Réalisation : Guillaume DELACROIX Formateur
AFPA

26 août 2024

Afpa

```

        <button type="submit" className="btn btn-primary">Add Product</button>
      </form>
    </div>
  );
}
function generateCategoryId() {
  const timestamp = Math.floor(new Date().getTime() / 1000).toString(16);
  const machineIdentifier = Math.floor(Math.random() * 0xffffffff)
    .toString(16)
    .padStart(6, "0");
  const processIdentifier = Math.floor(Math.random() * 0xffff)
    .toString(16)
    .padStart(4, "0");
  const counter = Math.floor(Math.random() * 0xffffffff)
    .toString(16)
    .padStart(6, "0");

  return timestamp + machineIdentifier + processIdentifier + counter;
}

export default AddProduct;

```

Étape 7 : Créer le Composant `ProductList`

Ce composant affichera la liste des produits.

Chemin : `src/components/ProductList.jsx`

```

import React, { useState, useEffect } from 'react';
import { getProducts } from '../services/productService';

function ProductList() {
  const [products, setProducts] = useState([]);

  useEffect(() => {
    getProducts().then(setProducts);
  }, []);

  return (
    <div className="container mt-5">
      <h2>Product List</h2>
      <ul className="list-group">
        {products.map(product => (
          <li key={product._id} className="list-group-item">
            <strong>{product.name}</strong> - ${product.price}
            <span className="badge bg-secondary ms-2">{product.category.name}</span>
          </li>
        ))}
      </ul>
    </div>
  );
}

export default ProductList;

```

React

Réalisation : Guillaume DELACROIX Formateur
AFPA

26 août 2024

Afpa

Étape 8 : Intégrer les Composants dans `App.jsx`

Mettez à jour `App.jsx` pour inclure les routes vers vos composants.

Chemin : `src/App.jsx`

```
import React from 'react';
import { BrowserRouter as Router, Route, Routes, Link } from 'react-router-dom';
import ProductList from './components/ProductList';
import AddProduct from './components/AddProduct';

function App() {
  return (
    <Router>
      <div className="container mt-5">
        <nav className="mb-4">
          <Link to="/" className="btn btn-primary me-2">Product List</Link>
          <Link to="/add-product" className="btn btn-secondary">Add Product</Link>
        </nav>

        <Routes>
          <Route path="/" element={<ProductList />} />
          <Route path="/add-product" element={<AddProduct />} />
        </Routes>
      </div>
    </Router>
  );
}

export default App;
```

Étape 9 : Lancer l'Application

Enfin, pour tester votre application :

```
npm run dev
```

Accédez à `http://localhost:3000` pour voir votre application en action. Vous pouvez désormais ajouter des produits en sélectionnant une catégorie existante ou en en créant une nouvelle directement via l'interface utilisateur. Le tout est stylé avec Bootstrap pour une meilleure présentation.