

TP Mini-Projet React

Application de gestion d'articles

Objectifs pédagogiques

Ce mini-projet permet de mobiliser l'ensemble des compétences vues :

- composants et props
- state et événements
- effets et appels API
- listes et rendu conditionnel
- formulaires contrôlés
- navigation avec React Router

Présentation du projet

L'application permet :

- d'afficher une liste d'articles depuis une API
- de naviguer entre plusieurs pages
- d'ajouter un article via un formulaire

1 Initialisation du projet

Les commandes suivantes permettent de créer et lancer le projet.

```
npm create vite@latest mini-projet-react
cd mini-projet-react
npm install
npm install react-router-dom
npm run dev
```

2 Structure du projet

```
src/  
  components/  
    ArticleCard.jsx  
  pages/  
    Home.jsx  
    Articles.jsx  
    AddArticle.jsx  
  services/  
    api.js  
  App.jsx  
  main.jsx
```

3 Mise en place de la navigation

Dans `main.jsx`, l'application est enveloppée par le routeur.

```
import { BrowserRouter } from "react-router-dom";  
import App from "../App";
```

```
<BrowserRouter>  
  <App />  
</BrowserRouter>
```

Dans `App.jsx` :

```
import { Routes, Route, Link } from "react-router-dom";  
import Home from "../pages/Home";  
import Articles from "../pages/Articles";  
import AddArticle from "../pages/AddArticle";  
  
function App() {  
  return (  
    <>  
      <nav>  
        <Link to="/">Accueil</Link>  
        <Link to="/articles">Articles</Link>  
        <Link to="/add">Ajouter</Link>  
      </nav>  
  
      <Routes>  
        <Route path="/" element={<Home />} />  
        <Route path="/articles" element={<Articles />} />  
        <Route path="/add" element={<AddArticle />} />  
      </Routes>  
    </>  
  );  
}
```

```
export default App;
```

4 Service d'accès à l'API

Les appels réseau sont centralisés dans un service.

```
export function getArticles() {  
  return fetch("https://jsonplaceholder.typicode.com/posts")  
    .then(res => res.json());  
}
```

5 Affichage des articles

Dans `Articles.jsx` :

```
import { useEffect, useState } from "react";  
import { getArticles } from "../services/api";  
import ArticleCard from "../components/ArticleCard";  
  
function Articles() {  
  const [articles, setArticles] = useState([]);  
  
  useEffect(() => {  
    getArticles().then(data => setArticles(data));  
  }, []);  
  
  return (  
    <div>  
      <h1>Articles</h1>  
      {articles.map(article => (  
        <ArticleCard  
          key={article.id}  
          titre={article.title}  
          contenu={article.body}  
        />  
      ))}  
    </div>  
  );  
}  
  
export default Articles;
```

6 Composant réutilisable

Le composant `ArticleCard` affiche un article.

```
function ArticleCard({ titre, contenu }) {  
  return (  
    <div>  
      <h2>{titre}</h2>  
      <p>{contenu}</p>  
    </div>  
  );  
}  
  
export default ArticleCard;
```

7 Formulaire d'ajout

Dans `AddArticle.jsx` :

```
import { useState } from "react";  
  
function AddArticle() {  
  const [titre, setTitre] = useState("");  
  const [contenu, setContenu] = useState("");  
  
  const handleSubmit = (e) => {  
    e.preventDefault();  
    console.log({ titre, contenu });  
  };  
  
  return (  
    <form onSubmit={handleSubmit}>  
      <input  
        value={titre}  
        onChange={(e) => setTitre(e.target.value)}  
        placeholder="Titre"  
      />  
      <textarea  
        value={contenu}  
        onChange={(e) => setContenu(e.target.value)}  
        placeholder="Contenu"  
      />  
      <button type="submit">Ajouter</button>  
    </form>  
  );  
}  
  
export default AddArticle;
```

Conclusion pédagogique

Ce mini-projet constitue une synthèse complète du parcours React :

- architecture claire
- logique maîtrisée
- composants réutilisables
- navigation fonctionnelle

Base solide pour un projet React réel.