

# TP React 10

## Architecture et bonnes pratiques

### Objectifs pédagogiques

À la fin de ce TP, vous serez capable de :

- structurer proprement une application React
- créer des composants réutilisables
- séparer la logique et l'affichage
- appliquer les bonnes pratiques React

### Introduction

À mesure qu'une application React grandit, il devient essentiel d'organiser le code pour qu'il reste :

- lisible
- maintenable
- évolutif

Une bonne architecture facilite le travail en équipe et la maintenance du projet.

### 1 Structure de projet recommandée

Une organisation claire des fichiers est indispensable.

```
src/
  components/
    Header.jsx
    Footer.jsx
    ArticleCard.jsx
  pages/
    Home.jsx
    Articles.jsx
    About.jsx
  services/
    api.js
  App.jsx
  main.jsx
```

Chaque dossier a une responsabilité claire.

## 2 Composants réutilisables

Un composant doit idéalement :

- avoir un rôle précis
- être réutilisable
- recevoir ses données via des props

Exemple : ArticleCard.jsx

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

## 3 Séparation logique / affichage

Il est recommandé de séparer :

- la logique (state, appels API)
- l'affichage (JSX)

Exemple :

```
function ArticlesContainer() {  
  const [articles, setArticles] = useState([]);  
  
  return <ArticlesList articles={articles} />;  
}
```

## 4 Centralisation des appels API

Les appels réseau doivent être regroupés dans un service.

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

Centraliser les appels API évite la duplication du code.

## 5 Bonnes pratiques générales

- nommer clairement les composants
- éviter les composants trop longs
- limiter le nombre de states
- découper l'interface en petites briques

Si un composant dépasse 200 lignes, il est souvent temps de le découper.

## 6 Préparer un projet réel

Avant de démarrer un projet React réel :

- définir les pages
- identifier les composants réutilisables
- organiser les dossiers
- planifier les appels API

## Conclusion générale du parcours

Au fil des TP, vous avez appris à :

- créer et organiser des composants React
- gérer le state et les effets
- afficher des listes dynamiques
- gérer des formulaires contrôlés
- communiquer avec une API
- naviguer entre plusieurs pages
- structurer une application complète

**Vous disposez désormais des bases solides pour développer une application React professionnelle.**