

Architecture React/Vite - Plateforme Porsche

📋 Table des matières

1. [Analyse de l'architecture Node existante](#)
2. [Architecture React proposée](#)
3. [Structure détaillée des dossiers](#)
4. [Bonnes pratiques](#)
5. [Exemples de code](#)

🔍 Analyse de l'architecture Node existante {#analyse-backend}

Structure Backend actuelle

```
Node/
  ├── server.js          # Point d'entrée, configuration Express
  ├── controllers/       # Logique métier (19 contrôleurs)
  ├── models/             # Schémas MongoDB (18 modèles)
  ├── routes/             # Routes API REST
  ├── middlewares/        # Auth, validation, sécurité
  ├── utils/              # Utilitaires, constants, helpers
  ├── validations/        # Validations Joi/Express-validator
  ├── db/                 # Configuration base de données
  └── uploads/            # Fichiers uploadés
```

Entités principales identifiées

👤 Gestion des utilisateurs

- **user** - Utilisateurs (client, conseiller, responsable, admin)
- Rôles: client, conseillere, responsable, admin

🚗 Gestion des voitures

- **voiture** - Modèle général (911, Cayman, Cayenne)
- **model_porsche** - Variantes (Carrera, Turbo, GTS)
- **model_porsche_actuel** - Voitures d'occasion
- **photo_voiture / photo_porsche / photo_voiture_actuel** - Photos

🎨 Personnalisation

- **couleur_exterieur** - Couleurs extérieures
- **couleur_interieur** - Couleurs intérieures
- **siege** - Types de sièges

- **taille_jante** - Tailles de jantes
- **package** - Packs d'options

🛍️ Accessoires

- **accesoire** - Accessoires disponibles
- **couleur_accesoire** - Couleurs d'accessoires
- **photo_accesoire** - Photos d'accessoires

💳 Commandes & Paiements

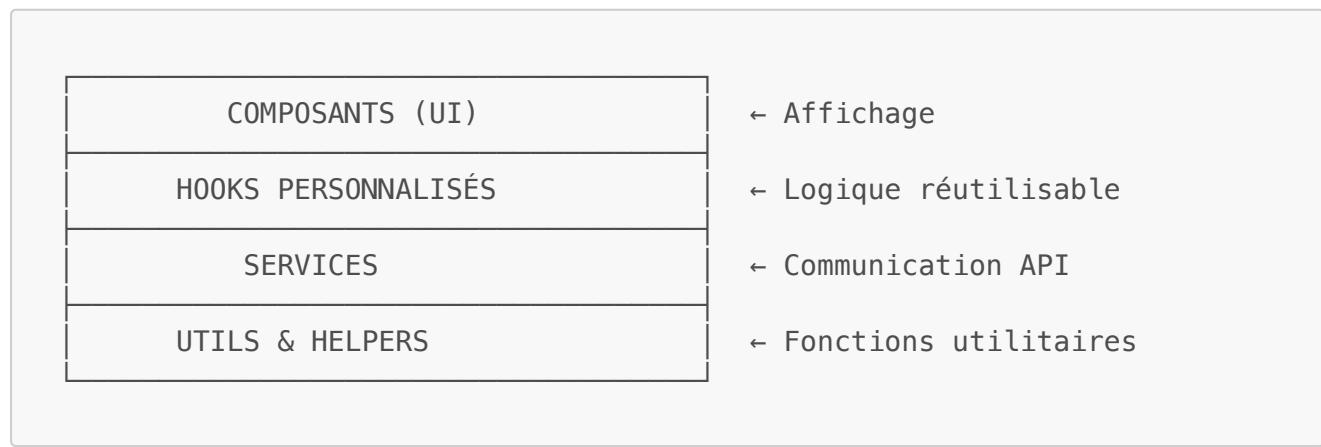
- **Commande** - Commandes/Panier
- **ligneCommande** - Lignes de commande
- **reservation** - Réservations
- **payment** - Paiements Stripe

🎯 Architecture React proposée {#architecture-react}

Principes SOLID appliqués

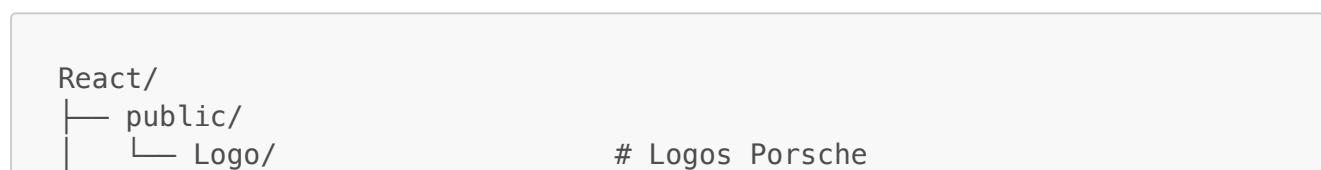
1. **Single Responsibility** - Chaque composant a une seule responsabilité
2. **Open/Closed** - Composants extensibles via props
3. **Liskov Substitution** - Composants réutilisables
4. **Interface Segregation** - Props spécifiques
5. **Dependency Inversion** - Services abstraits

Architecture en couches



📁 Structure détaillée des dossiers {#structure-dossiers}

Vue d'ensemble



```
src/
  └── main.jsx          # Point d'entrée React
  └── App.jsx           # Composant racine
  └── App.css           # Styles globaux
  └── index.css         # Reset CSS

  └── config/           # ⚙ Configuration
    ├── api.js           # Configuration Axios
    ├── routes.js        # Routes de l'application
    └── constantes.js    # Constantes globales

  └── services/          # 🚀 Services API
    └── api/
      ├── authService.js
      ├── voitureService.js
      ├── accesoireService.js
      ├── commandeService.js
      ├── personnalisationService.js
      └── paiementService.js
    └── index.js          # Export centralisé

  └── hooks/             # 🔗 Hooks personnalisés
    ├── useAuth.js        # Authentification
    ├── useVoiture.js     # Gestion voitures
    ├── usePanier.js       # Gestion panier
    ├── useLocalStorage.js # Storage local
    └── index.js

  └── contextes/          # 🌐 Context API
    ├── AuthContexte.jsx
    ├── PanierContexte.jsx
    ├── ConfigurateurContexte.jsx
    └── index.js

  └── pages/              # 📄 Pages principales
    └── Accueil/
      ├── Accueil.jsx
      └── Accueil.css

    └── Authentification/
      ├── Connexion.jsx
      ├── Connexion.css
      ├── Inscription.jsx
      └── Inscription.css

    └── Catalogue/
      ├── CatalogueVoitures.jsx
      ├── CatalogueVoitures.css
      ├── DetailVoiture.jsx
      └── DetailVoiture.css

    └── Configurateur/
      ├── Configurateur.jsx
```

```
    ├── Configurateur.css
    ├── EtapeCouleur.jsx
    ├── EtapeJantes.jsx
    ├── EtapeSieges.jsx
    ├── EtapePackages.jsx
    └── Recapitulatif.jsx

    ├── Accessoires/
    │   ├── CatalogueAccessoires.jsx
    │   ├── CatalogueAccessoires.css
    │   ├── DetailAccessoire.jsx
    │   └── DetailAccessoire.css

    ├── Panier/
    │   ├── Panier.jsx
    │   └── Panier.css

    ├── Commande/
    │   ├── Commande.jsx
    │   ├── Commande.css
    │   ├── Paiement.jsx
    │   ├── ConfirmationCommande.jsx
    │   └── HistoriqueCommandes.jsx

    ├── Profil/
    │   ├── MonProfil.jsx
    │   ├── MonProfil.css
    │   ├── MesReservations.jsx
    │   └── MesCommandes.jsx

    ├── Administration/
    │   ├── TableauDeBord.jsx
    │   ├── GestionVoitures.jsx
    │   ├── GestionAccessoires.jsx
    │   ├── GestionUtilisateurs.jsx
    │   ├── GestionCommandes.jsx
    │   └── Statistiques.jsx

    ├── Erreur/
    │   ├── Page404.jsx
    │   └── Page404.css

    └── composants/
        ├── communs/          # 🌟 Composants réutilisables
        │   ├── Bouton/
        │   │   ├── Bouton.jsx
        │   │   └── Bouton.css
        |
        └── Carte/            # Composants génériques
            ├── Carte.jsx
            └── Carte.css
```

```
    └── Chargement/
        ├── Chargement.jsx
        └── Chargement.css

    └── Modale/
        ├── Modale.jsx
        └── Modale.css

    └── Formulaire/
        ├── ChampTexte.jsx
        ├── ChampTexte.css
        ├── ChampEmail.jsx
        ├── ChampMotDePasse.jsx
        ├── SelectMenu.jsx
        └── BoutonSoumission.jsx

    └── Notification/
        ├── Notification.jsx
        └── Notification.css

    └── Pagination/
        ├── Pagination.jsx
        └── Pagination.css

    └── Tableau/
        ├── Tableau.jsx
        └── Tableau.css

    └── layout/          # Structure de page
        └── EnTete/
            ├── EnTete.jsx
            └── EnTete.css

        └── Navigation/
            ├── Navigation.jsx
            └── Navigation.css

        └── PiedDePage/
            ├── PiedDePage.jsx
            └── PiedDePage.css

        └── MiseEnPage/
            ├── MiseEnPage.jsx
            └── MiseEnPage.css

    └── voiture/         # Composants voitures
        └── CarteVoiture/
            ├── CarteVoiture.jsx
            └── CarteVoiture.css

        └── ListeVoitures/
            ├── ListeVoitures.jsx
            └── ListeVoitures.css
```

```
    └── GaleriePhotos/
        ├── GaleriePhotos.jsx
        └── GaleriePhotos.css

    └── FichesTechniques/
        ├── FichesTechniques.jsx
        └── FichesTechniques.css

    └── FiltreVoiture/
        ├── FiltreVoiture.jsx
        └── FiltreVoiture.css

    └── configIBUTEUR/      # Composants configuration
        └── SelecteurCouleur/
            ├── SelecteurCouleur.jsx
            └── SelecteurCouleur.css

        └── SelecteurJantes/
            ├── SelecteurJantes.jsx
            └── SelecteurJantes.css

        └── SelecteurSieges/
            ├── SelecteurSieges.jsx
            └── SelecteurSieges.css

        └── SelecteurPackage/
            ├── SelecteurPackage.jsx
            └── SelecteurPackage.css

        └── RecapitulatifConfig/
            ├── RecapitulatifConfig.jsx
            └── RecapitulatifConfig.css

    └── accessoire/          # Composants accessoires
        └── CarteAccessoire/
            ├── CarteAccessoire.jsx
            └── CarteAccessoire.css

        └── ListeAccessoires/
            ├── ListeAccessoires.jsx
            └── ListeAccessoires.css

    └── panier/              # Composants panier
        └── ArticlePanier/
            ├── ArticlePanier.jsx
            └── ArticlePanier.css

        └── ResUMEpanier/
            ├── ResUMEpanier.jsx
            └── ResUMEpanier.css

        └── IconePanier/
```

```

    └── IcônePanier.jsx
    └── IcônePanier.css

    └── commande/          # Composants commande
        └── FormulaireCommande/
            ├── FormulaireCommande.jsx
            └── FormulaireCommande.css

        └── RecapitulatifCommande/
            ├── RecapitulatifCommande.jsx
            └── RecapitulatifCommande.css

        └── StatutCommande/
            ├── StatutCommande.jsx
            └── StatutCommande.css

    └── paiement/          # Composants paiement
        └── FormulairePaiement/
            ├── FormulairePaiement.jsx
            └── FormulairePaiement.css

        └── ConfirmationPaiement/
            ├── ConfirmationPaiement.jsx
            └── ConfirmationPaiement.css

    └── admin/              # Composants admin
        └── TableauStatistiques/
            ├── TableauStatistiques.jsx
            └── TableauStatistiques.css

        └── FormulaireVoiture/
            ├── FormulaireVoiture.jsx
            └── FormulaireVoiture.css

        └── GestionPhotos/
            ├── GestionPhotos.jsx
            └── GestionPhotos.css

    └── protection/         # Routes protégées
        ├── RoutePrivee.jsx
        ├── RouteAdmin.jsx
        └── RouteConseiller.jsx

    └── utils/               # 🛠 Utilitaires
        ├── formatage.js      # Formatage dates, prix
        ├── validation.js      # Validation formulaires
        ├── calculPrix.js      # Calculs de prix
        ├── gestionErreurs.js  # Gestion erreurs
        ├── constantes.js      # Constantes
        └── helpers.js         # Fonctions helpers

    └── styles/              # 🎨 Styles globaux
        └── variables.css      # Variables CSS (couleurs, fonts)

```

```

    └── reset.css          # Reset CSS
    └── animations.css    # Animations
    └── mixins.css        # Mixins/utilitaires

    └── assets/            # 📁 Assets statiques
        └── images/
        └── icons/
        └── fonts/

    └── .env.development   # Variables d'environnement dev
    └── .env.production    # Variables d'environnement prod
    └── .eslintrc.js       # Configuration ESLint
    └── .prettierrc        # Configuration Prettier
    └── vite.config.js     # Configuration Vite
    └── package.json        # Dépendances

```

✨ Bonnes pratiques {#bonnes-pratiques}

1. 📁 Organisation des composants

Principe : Un dossier = Un composant

```

Bouton/
└── Bouton.jsx      # Composant
└── Bouton.css      # Styles
└── index.js        # Export (optionnel)

```

Export nommé vs export par défaut

```

// ✅ RECOMMANDÉ : Export par défaut
// Bouton.jsx
export default function Bouton({ texte, onClick }) {
  return <button onClick={onClick}>{texte}</button>;
}

// Import
import Bouton from "./Bouton/Bouton";

// ✅ ALTERNATIF : Export nommé + index.js
// Bouton.jsx
export function Bouton({ texte, onClick }) {
  return <button onClick={onClick}>{texte}</button>;
}

// index.js
export { Bouton } from "./Bouton";

```

```
// Import
import { Bouton } from "./Bouton";
```

2. ⚙ Séparation des responsabilités

Composants de présentation (Dumb Components)

```
// CarteVoiture.jsx - Affichage uniquement
export default function CarteVoiture({ voiture, onSelection }) {
  return (
    <div className="carte-voiture">
      <img src={voiture.photo} alt={voiture.nom} />
      <h3>{voiture.nom}</h3>
      <p>{voiture.prix}€</p>
      <button onClick={() =>
        onSelection(voiture.id)}>Configurer</button>
    </div>
  );
}
```

Composants conteneurs (Smart Components)

```
// CatalogueVoitures.jsx - Logique métier
import { useState, useEffect } from "react";
import { voitureService } from "@/services";
import CarteVoiture from
"@/composants/voiture/CarteVoiture/CarteVoiture";

export default function CatalogueVoitures() {
  const [voitures, setVoitures] = useState([]);
  const [chargement, setChangement] = useState(true);

  useEffect(() => {
    const chargerVoitures = async () => {
      try {
        const data = await voitureService.obtenirTous();
        setVoitures(data);
      } catch (erreur) {
        console.error(erreur);
      } finally {
        setChangement(false);
      }
    };
    chargerVoitures();
  }, []);
}
```

```

const gererSelection = (id) => {
    // Logique de navigation
};

if (chargement) return <Chargement />

return (
    <div className="catalogue">
        {voitures.map((voiture) => (
            <CarteVoiture
                key={voiture.id}
                voiture={voiture}
                onSelection={gererSelection}
            />
        )))
    </div>
);
}

```

3. ⚡ Hooks personnalisés

Réutilisation de la logique

```

// hooks/useAuth.js
import { useState, useEffect, useContext } from "react";
import { AuthContexte } from "@/contextes/AuthContexte";
import { authService } from "@/services";

export function useAuth() {
    const contexte = useContext(AuthContexte);

    if (!contexte) {
        throw new Error("useAuth doit être utilisé dansAuthProvider");
    }

    const connexion = async (email, motDePasse) => {
        try {
            const data = await authService.connexion(email, motDePasse);
            contexte.setUtilisateur(data.user);
            localStorage.setItem("token", data.token);
            return { success: true };
        } catch (erreur) {
            return { success: false, erreur: erreur.message };
        }
    };

    const deconnexion = () => {
        contexte.setUtilisateur(null);
        localStorage.removeItem("token");
    };
}

```

```

};

return {
  utilisateur: contexte.utilisateur,
  estConnecte: !!contexte.utilisateur,
  connexion,
  deconnexion,
};
}

```

4. Context API pour état global

```

// contextes/PanierContexte.jsx
import { createContext, useState, useEffect } from "react";

export const PanierContexte = createContext();

export function PanierProvider({ children }) {
  const [articles, setArticles] = useState([]);

  // Charger depuis localStorage au montage
  useEffect(() => {
    const panierSauvegarde = localStorage.getItem("panier");
    if (panierSauvegarde) {
      setArticles(JSON.parse(panierSauvegarde));
    }
  }, []);

  // Sauvegarder dans localStorage à chaque changement
  useEffect(() => {
    localStorage.setItem("panier", JSON.stringify(articles));
  }, [articles]);

  const ajouterArticle = (article) => {
    setArticles((prev) => [...prev, article]);
  };

  const retirerArticle = (id) => {
    setArticles((prev) => prev.filter((article) => article.id !== id));
  };

  const viderPanier = () => {
    setArticles([]);
  };

  const totalPanier = articles.reduce((acc, article) => {
    return acc + article.prix;
  }, 0);

  const valeur = {

```

```

        articles,
        ajouterArticle,
        retirerArticle,
        viderPanier,
        totalPanier,
        nombreArticles: articles.length,
    };

    return (
        <PanierContexte.Provider value={valeur}>{children}
    </PanierContexte.Provider>
);
}

```

5. 🚗 Services API

```

// services/api/voitureService.js
import api from "@/config/api";

export const voitureService = {
    // Obtenir toutes les voitures
    obtenirTous: async () => {
        const response = await api.get("/voiture");
        return response.data;
    },

    // Obtenir une voiture par ID
    obtenirParId: async (id) => {
        const response = await api.get(`voiture/${id}`);
        return response.data;
    },

    // Créer une nouvelle voiture (admin)
    creer: async (donnees) => {
        const response = await api.post("/voiture", donnees);
        return response.data;
    },

    // Mettre à jour une voiture (admin)
    mettreAJour: async (id, donnees) => {
        const response = await api.put(`voiture/${id}`, donnees);
        return response.data;
    },

    // Supprimer une voiture (admin)
    supprimer: async (id) => {
        const response = await api.delete(`voiture/${id}`);
        return response.data;
    },
}

```

```
// Filtrer par type (neuve/occasion)
filtrerParType: async (type) => {
    const response = await api.get(`/voiture?type_voiture=${type}`);
    return response.data;
},
};
```

6. Configuration API (Axios)

```
// config/api.js
import axios from "axios";

const api = axios.create({
    baseURL: import.meta.env.VITE_API_URL || "http://localhost:3000",
    timeout: 10000,
    headers: {
        "Content-Type": "application/json",
    },
});

// Intercepteur pour ajouter le token
api.interceptors.request.use(
    (config) => {
        const token = localStorage.getItem("token");
        if (token) {
            config.headers.Authorization = `Bearer ${token}`;
        }
        return config;
    },
    (erreur) => {
        return Promise.reject(erreur);
    }
);

// Intercepteur pour gérer les erreurs
api.interceptors.response.use(
    (response) => response,
    (erreur) => {
        if (erreur.response?.status === 401) {
            // Déconnexion automatique si non autorisé
            localStorage.removeItem("token");
            window.location.href = "/connexion";
        }
        return Promise.reject(erreur);
    }
);

export default api;
```

7. 🛡️ Routes protégées

```
// composants/protection/RoutePrivee.jsx
import { Navigate } from 'react-router-dom';
import { useAuth } from '@/hooks/useAuth';

export default function RoutePrivee({ children }) {
  const { estConnecte } = useAuth();

  if (!estConnecte) {
    return <Navigate to="/connexion" replace />;
  }

  return children;
}

// composants/protection/RouteAdmin.jsx
import { Navigate } from 'react-router-dom';
import { useAuth } from '@/hooks/useAuth';

export default function RouteAdmin({ children }) {
  const { utilisateur, estConnecte } = useAuth();

  if (!estConnecte) {
    return <Navigate to="/connexion" replace />;
  }

  if (utilisateur?.role !== 'admin') {
    return <Navigate to="/" replace />;
  }

  return children;
}
```

8. 🎨 Variables CSS

```
/* styles/variables.css */
:root {
  /* Couleurs Porsche */
  --couleur-principale: #000000;
  --couleur-secondaire: #d5001c;
  --couleur-accent: #c0a062;

  --couleur-texte: #333333;
  --couleur-texte-clair: #666666;
  --couleur-fond: #ffffff;
  --couleur-fond-gris: #f5f5f5;

  /* Espacements */
}
```

```

--espace-xs: 0.25rem;
--espace-sm: 0.5rem;
--espace-md: 1rem;
--espace-lg: 1.5rem;
--espace-xl: 2rem;
--espace-xxl: 3rem;

/* Typographie */
--font-principale: "Porsche Next", Arial, sans-serif;
--font-taille-xs: 0.75rem;
--font-taille-sm: 0.875rem;
--font-taille-base: 1rem;
--font-taille-lg: 1.125rem;
--font-taille-xl: 1.25rem;
--font-taille-xxl: 1.5rem;
--font-taille-titre: 2rem;

/* Ombres */
--ombre-sm: 0 1px 2px rgba(0, 0, 0, 0.05);
--ombre-md: 0 4px 6px rgba(0, 0, 0, 0.1);
--ombre-lg: 0 10px 15px rgba(0, 0, 0, 0.1);

/* Transitions */
--transition-rapide: 150ms ease-in-out;
--transition-normale: 300ms ease-in-out;
--transition-lente: 500ms ease-in-out;

/* Bordures */
--rayon-bordure-sm: 4px;
--rayon-bordure-md: 8px;
--rayon-bordure-lg: 12px;
}

```

9. ✎ Validation de formulaires

```

// utils/validation.js
export const validationEmail = (email) => {
  const regex = /^[^@\s]+@[^\s@]+\.\w+$/;
  return regex.test(email);
};

export const validationMotDePasse = (motDePasse) => {
  // Minimum 8 caractères, 1 majuscule, 1 minuscule, 1 chiffre
  const regex = /^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)[a-zA-Z\d]{8,}$/;
  return regex.test(motDePasse);
};

export const validationTelephone = (telephone) => {
  // Format français: 0X XX XX XX XX
  const regex = /^0[1-9](?:[\s.-]?\d{2}){4}$/;
}

```

```

    return regex.test(telephone);
};

export const validationCodePostal = (codePostal) => {
    // Format français: 5 chiffres
    const regex = /^d{5}$/;
    return regex.test(codePostal);
};

```

10. 💰 Formatage et calculs

```

// utils/formatage.js
export const formaterPrix = (prix) => {
    return new Intl.NumberFormat("fr-FR", {
        style: "currency",
        currency: "EUR",
        minimumFractionDigits: 0,
        maximumFractionDigits: 0,
    }).format(prix);
};

export const formaterDate = (date) => {
    return new Intl.DateTimeFormat("fr-FR", {
        year: "numeric",
        month: "long",
        day: "numeric",
    }).format(new Date(date));
};

export const formaterDateHeure = (date) => {
    return new Intl.DateTimeFormat("fr-FR", {
        year: "numeric",
        month: "long",
        day: "numeric",
        hour: "2-digit",
        minute: "2-digit",
    }).format(new Date(date));
};

```

```

// utils/calculPrix.js
export const calculerPrixTotal = (configuration) => {
    let total = configuration.prixBase || 0;

    // Ajouter options
    if (configuration.couleurExterieur) {
        total += configuration.couleurExterieur.prix || 0;
    }
}

```

```

if (configuration.couleurInterieur) {
    total += configuration.couleurInterieur.prix || 0;
}

if (configuration.jantes) {
    total += configuration.jantes.prix || 0;
}

if (configuration.sieges) {
    total += configuration.sieges.prix || 0;
}

if (configuration.package) {
    total += configuration.package.prix || 0;
}

// Ajouter accessoires
if (configuration.accessoires) {
    configuration.accessoires.forEach((acc) => {
        total += acc.prix * (acc.quantite || 1);
    });
}

return total;
};

```



Exemples de code {#exemples}

Exemple 1: Page de connexion complète

```

// pages/Authentification/Connexion.jsx
import { useState } from "react";
import { useNavigate } from "react-router-dom";
import { useAuth } from "@hooks/useAuth";
import ChampEmail from "@/composants/communs/Formulaire/ChampEmail";
import ChampMotDePasse from
"@/composants/communs/Formulaire/ChampMotDePasse";
import Bouton from "@/composants/communs/Bouton/Bouton";
import Notification from
"@/composants/communs/Notification/Notification";
import "./Connexion.css";

export default function Connexion() {
    const navigate = useNavigate();
    const { connexion } = useAuth();

    const [formulaire, setFormulaire] = useState({
        email: "",
        motDePasse: "",
    });
}

```

```

    });

const [erreur, setErreur] = useState("");
const [chargement, setChangement] = useState(false);

const gererChangement = (e) => {
  const { name, value } = e.target;
  setFormulaire((prev) => ({
    ...prev,
    [name]: value,
  }));
};

const gererSoumission = async (e) => {
  e.preventDefault();
  setErreur("");
  setChangement(true);

  const resultat = await connexion(formulaire.email,
  formulaire.motDePasse);

  setChangement(false);

  if (resultat.success) {
    navigate("/");
  } else {
    setErreur(resultat.erreur);
  }
};

return (
  <div className="connexion">
    <div className="connexion__conteneur">
      <h1 className="connexion__titre">Connexion</h1>

      {erreur && <Notification type="erreur" message={erreur} />}

      <form className="connexion__formulaire" onSubmit={gererSoumission}>
        <ChampEmail
          nom="email"
          valeur={formulaire.email}
          onChange={gererChangement}
          requis
        />

        <ChampMotDePasse
          nom="motDePasse"
          valeur={formulaire.motDePasse}
          onChange={gererChangement}
          requis
        />
    
```

```

        <Bouton
            type="submit"
            texte={chargement ? "Connexion..." : "Se connecter"}
            desactive={chargement}
            pleineLargeur
        />
    </form>

    <p className="connexion_lien">
        Pas encore de compte ? <a href="/inscription">S'inscrire</a>
    </p>
</div>
</div>
);
}

```

```

/* pages/Authentification/Connexion.css */
.connexion {
    min-height: 100vh;
    display: flex;
    align-items: center;
    justify-content: center;
    background-color: var(--couleur-fond-gris);
    padding: var(--espace-xl);
}

.connexion__conteneur {
    background: var(--couleur-fond);
    padding: var(--espace-xxl);
    border-radius: var(--rayon-bordure-lg);
    box-shadow: var(--ombre-lg);
    width: 100%;
    max-width: 400px;
}

.connexion__titre {
    font-size: var(--font-taille-titre);
    margin-bottom: var(--espace-xl);
    text-align: center;
    color: var(--couleur-principale);
}

.connexion__formulaire {
    display: flex;
    flex-direction: column;
    gap: var(--espace-lg);
}

.connexion__lien {
    margin-top: var(--espace-lg);
}

```

```

    text-align: center;
    color: var(--couleur-texte-clair);
}

.connexion_lien a {
    color: var(--couleur-secondaire);
    text-decoration: none;
    font-weight: 600;
    transition: color var(--transition-rapide);
}

.connexion_lien a:hover {
    color: var(--couleur-principale);
}

```

Exemple 2 : Composant Carte Voiture

```

// composants/voiture/CarteVoiture/CarteVoiture.jsx
import { useNavigate } from "react-router-dom";
import { formaterPrix } from "@utils/formatage";
import Bouton from "@/composants/communs/Bouton/Bouton";
import "./CarteVoiture.css";

export default function CarteVoiture({ voiture }) {
    const navigate = useNavigate();

    const photoUrl = voiture.photo_voiture?.[0]?.url || "/images/default-car.jpg";

    const gererConfiguration = () => {
        navigate(`/configurateur/${voiture._id}`);
    };

    const gererDetail = () => {
        navigate(`/catalogue/${voiture._id}`);
    };

    return (
        <div className="carte-voiture">
            <div className="carte-voiture__image-conteneur">
                <img
                    src={photoUrl}
                    alt={voiture.nom_model}
                    className="carte-voiture__image"
                />
                {voiture.type_voiture && (
                    <span className="carte-voiture__badge">Neuve</span>
                )}
            </div>
        </div>
    );
}

```

```

<div className="carte-voiture__contenu">
  <h3 className="carte-voiture__titre">{voiture.nom_model}</h3>

  <p className="carte-voiture__description">
    {voiture.description?.substring(0, 100)}...
  </p>

  <div className="carte-voiture__footer">
    <Bouton
      texte="Voir détails"
      onClick={gererDetail}
      variante="secondaire"
    />
    <Bouton
      texte="Configurer"
      onClick={gererConfiguration}
      variante=" primaire"
    />
  </div>
</div>
</div>
);
}

```

```

/* composants/voiture/CarteVoiture/CarteVoiture.css */
.carte-voiture {
  background: var(--couleur-fond);
  border-radius: var(--rayon-bordure-md);
  overflow: hidden;
  box-shadow: var(--ombre-md);
  transition: transform var(--transition-normale), box-shadow var(--transition-normale);
  height: 100%;
  display: flex;
  flex-direction: column;
}

.carte-voiture:hover {
  transform: translateY(-4px);
  box-shadow: var(--ombre-lg);
}

.carte-voiture__image-conteneur {
  position: relative;
  width: 100%;
  height: 250px;
  overflow: hidden;
  background-color: var(--couleur-fond-gris);
}

```

```
.carte-voiture__image {
  width: 100%;
  height: 100%;
  object-fit: cover;
  transition: transform var(--transition-lente);
}

.carte-voiture:hover .carte-voiture__image {
  transform: scale(1.05);
}

.carte-voiture__badge {
  position: absolute;
  top: var(--espace-md);
  right: var(--espace-md);
  background-color: var(--couleur-secondaire);
  color: var(--couleur-fond);
  padding: var(--espace-xs) var(--espace-md);
  border-radius: var(--rayon-bordure-sm);
  font-size: var(--font-taille-sm);
  font-weight: 600;
}

.carte-voiture__contenu {
  padding: var(--espace-lg);
  display: flex;
  flex-direction: column;
  gap: var(--espace-md);
  flex: 1;
}

.carte-voiture__titre {
  font-size: var(--font-taille-xl);
  color: var(--couleur-principale);
  margin: 0;
}

.carte-voiture__description {
  color: var(--couleur-texte-clair);
  font-size: var(--font-taille-sm);
  line-height: 1.5;
  margin: 0;
}

.carte-voiture__footer {
  display: flex;
  gap: var(--espace-md);
  margin-top: auto;
}
```

Exemple 3 : Hook personnalisé panier

```

// hooks/usePanier.js
import { useContext } from "react";
import { PanierContexte } from "@/contextes/PanierContexte";
import { calculerPrixTotal } from "@/utils/calculPrix";

export function usePanier() {
  const contexte = useContext(PanierContexte);

  if (!contexte) {
    throw new Error("usePanier doit être utilisé dans PanierProvider");
  }

  const ajouterVoiture = (voiture, configuration) => {
    const article = {
      id: Date.now(),
      type: "voiture",
      voiture,
      configuration,
      prix: calculerPrixTotal(configuration),
    };

    contexte.ajouterArticle(article);
  };

  const ajouterAccessoire = (accessoire, quantite = 1) => {
    const article = {
      id: Date.now(),
      type: "accessoire",
      accessoire,
      quantite,
      prix: accessoire.prix * quantite,
    };

    contexte.ajouterArticle(article);
  };

  const modifierQuantite = (id, quantite) => {
    const article = contexte.articles.find((a) => a.id === id);
    if (article && article.type === "accessoire") {
      const nouvelArticle = {
        ...article,
        quantite,
        prix: article.accessoire.prix * quantite,
      };
      contexte.retirerArticle(id);
      contexte.ajouterArticle(nouvelArticle);
    }
  };

  return {
    articles: contexte.articles,
  };
}

```

```

        ajouterVoiture,
        ajouterAccessoire,
        retirerArticle: contexte.retirerArticle,
        viderPanier: contexte.viderPanier,
        modifierQuantite,
        totalPanier: contexte.totalPanier,
        nombreArticles: contexte.nombreArticles,
    };
}

```

Exemple 4 : Service complet

```

// services/api/personnalisationService.js
import api from "@/config/api";

export const personnalisationService = {
    // Couleurs extérieures
    couleursExterieures: {
        obtenirTous: async () => {
            const response = await api.get("/couleur_exterieur");
            return response.data;
        },
        obtenirParId: async (id) => {
            const response = await api.get(`/couleur_exterieur/${id}`);
            return response.data;
        },
    },

    // Couleurs intérieures
    couleursInterieures: {
        obtenirTous: async () => {
            const response = await api.get("/couleur_interieur");
            return response.data;
        },
        obtenirParId: async (id) => {
            const response = await api.get(`/couleur_interieur/${id}`);
            return response.data;
        },
    },

    // Jantes
    jantes: {
        obtenirTous: async () => {
            const response = await api.get("/taille_jante");
            return response.data;
        },
        obtenirParId: async (id) => {
            const response = await api.get(`/taille_jante/${id}`);
            return response.data;
        },
    },
}

```

```

},
// Sièges
sieges: {
    obtenerTous: async () => {
        const response = await api.get("/siege");
        return response.data;
    },
    obtenerParId: async (id) => {
        const response = await api.get(`/siege/${id}`);
        return response.data;
    },
},
// Packages
packages: {
    obtenerTous: async () => {
        const response = await api.get("/package");
        return response.data;
    },
    obtenerParId: async (id) => {
        const response = await api.get(`/package/${id}`);
        return response.data;
    },
},
};


```

🎓 Conseils pédagogiques

Pour bien démarrer

1. **Commencez simple** : Ne créez pas tous les dossiers d'un coup
2. **Ajoutez au fur et à mesure** : Créez les composants quand vous en avez besoin
3. **Réutilisez** : Un composant bien fait peut être utilisé partout
4. **Testez régulièrement** : Vérifiez que tout fonctionne après chaque ajout
5. **Commentez votre code** : Expliquez ce que fait votre code

Ordre de développement recommandé

1. **Configuration de base** (config/, services/)
2. **Contextes globaux** (AuthContexte, PanierContexte)
3. **Composants communs** (Bouton, Chargement, etc.)
4. **Layout** (EnTete, Navigation, PiedDePage)
5. **Pages principales** (Accueil, Catalogue)
6. **Fonctionnalités complexes** (Configurateur, Panier)
7. **Administration** (si nécessaire)

Commandes utiles

```
# Créer un nouveau composant
mkdir -p src/composants/communs/Bouton
touch src/composants/communs/Bouton/Bouton.jsx
touch src/composants/communs/Bouton/Bouton.css

# Installer les dépendances essentielles
npm install react-router-dom axios

# Lancer le serveur de développement
npm run dev

# Builder pour la production
npm run build
```

Ressources complémentaires

Documentation officielle

- **React** : <https://react.dev>
- **Vite** : <https://vitejs.dev>
- **React Router** : <https://reactrouter.com>
- **Axios** : <https://axios-http.com>

Bonnes pratiques

- Principes SOLID en JavaScript
- Méthodologie Agile (Scrum, Kanban)
- Git Flow pour la gestion des branches
- Convention de nommage : camelCase pour JS, kebab-case pour CSS

Checklist de démarrage

- Configurer les variables d'environnement (.env)
- Installer les dépendances (npm install)
- Créer la structure de base (dossiers principaux)
- Configurer Axios (config/api.js)
- Créer les contextes (Auth, Panier)
- Développer les composants communs
- Créer le layout principal
- Implémenter l'authentification
- Développer le catalogue de voitures
- Créer le configurateur
- Implémenter le panier
- Intégrer le paiement
- Tester l'application

- Builder pour la production
-

Bon courage pour votre projet !  

N'hésitez pas à adapter cette architecture selon vos besoins spécifiques. L'important est de rester cohérent et organisé.