

# TP 03 : Application de gestion de voitures

- Backend : Laravel 10+
- Frontend : React 18+
- Base de données : MySQL ou SQLite
- API : REST (JSON)

## Partie 1 : Configuration du projet Laravel

### 1.1. Créer un projet Laravel

```
composer create-project laravel/laravel gestion-voitures-back
cd gestion-voitures-back
```

### 1.2. Configurer la base de données

Dans le fichier `.env` : (Il faut s'assurer que la base de données existe (phpMyAdmin, MySQL, etc.)

```
DB_DATABASE=voitures
DB_USERNAME=root
DB_PASSWORD=
```

### 1.3. Créer le modèle et la migration

```
php artisan make:model Voiture -m
```

Dans `database/migrations/xxxx_create_voitures_table.php` :

```
public function up()
{
    Schema::create('voitures', function (Blueprint $table) {
        $table->id();
        $table->string('marque');
        $table->string('modele');
        $table->year('annee');
        $table->string('couleur');
        $table->timestamps();
    });
}
```

Puis : `php artisan migrate` .

### 1.4. Créer le contrôleur API :

```
php artisan make:controller Api/VoitureController --api
```

### 1.5. Ajouter les routes API

Dans `routes/api.php` :

```
use App\Http\Controllers\Api\VoitureController;
Route::apiResource('voitures', VoitureController::class);
```

## 1.6. Implémenter le CRUD dans le contrôleur

Dans VoitureController.php :

```
use App\Models\Voiture;
use Illuminate\Http\Request;

class VoitureController extends Controller
{
    public function index() {
        return Voiture::all(); }

    public function store(Request $request) {
        $validated = $request->validate([
            'marque' => 'required|string|max:255',
            'modele' => 'required|string|max:255',
            'annee' => 'required|integer|min:1900|max:'.date('Y'),
            'couleur' => 'required|string|max:50',
        ]);

        return Voiture::create($validated); }

    public function show(Voiture $voiture) {
        return $voiture; }

    public function update(Request $request, Voiture $voiture) {
        $validated = $request->validate([
            'marque' => 'required|string|max:255',
            'modele' => 'required|string|max:255',
            'annee' => 'required|integer|min:1900|max:'.date('Y'),
            'couleur' => 'required|string|max:50',
        ]);

        $voiture->update($validated);
        return $voiture; }

    public function destroy(Voiture $voiture) {
        $voiture->delete();
        return response()->noContent(); }
}
```

## 1.7. Activer CORS

Dans app/Http/Middleware/HandleCors.php, ou installer fruitcake/laravel-cors si nécessaire.

Dans config/cors.php, mets :

```
'paths' => ['api/*'],
'allowed_origins' => ['http://localhost:3000'], // React
'allowed_methods' => ['*'],
'allowed_headers' => ['*'],
```

## Partie 2 : Configuration du projet React

### 2.1. Créer un projet React

```
npx create-react-app gestion-voitures-front  
cd gestion-voitures-front  
npm install axios react-router-dom
```

### 2.2. Arborescence

```
src/  
├── components/  
│   ├── VoitureList.js  
│   ├── VoitureForm.js  
│   └── VoitureEdit.js  
├── App.js  
└── api.js
```

### 2.3. Fichier api.js

```
import axios from 'axios';  
export default axios.create({baseUrl: 'http://localhost:8000/api',});
```

### 2.4. Composant VoitureList.js

```
import React, { useEffect, useState } from 'react';  
import api from '../api';  
  
function VoitureList() {  
  const [voitures, setVoitures] = useState([]);  
  useEffect(() => { api.get('/voitures').then(res => setVoitures(res.data)); }, []);  
  
  const deleteVoiture = id => {  
    api.delete(`/voitures/${id}`).then(() => {  
      setVoitures(voitures.filter(v => v.id !== id));  
    });  
  };  
  
  return (  
    <div> <h2>Liste des voitures</h2>  
    <a href="/ajouter">Ajouter une voiture</a>  
    <ul> {voitures.map(voiture => (  
      <li key={voiture.id}>  
        {voiture.marque} - {voiture.modele} ({voiture.annee})  
        [{voiture.couleur}]  
        <a href={`/modifier/${voiture.id}`}>Modifier</a>  
        <button onClick={() => deleteVoiture(voiture.id)}>Supprimer</button>  
      </li>  
    ) ) }  
    </ul> </div>  
  );  
}  
  
export default VoitureList;
```

## 2.5. Composant VoitureForm.js

```
import React, { useState } from 'react';
import api from '../api';
import { useNavigate } from 'react-router-dom';

function VoitureForm() {
  const navigate = useNavigate();
  const [form, setForm] = useState({
    marque: '',
    modele: '',
    annee: '',
    couleur: '', });

  const handleChange = e => {setForm({ ...form, [e.target.name]: e.target.value });};

  const handleSubmit = e => {
    e.preventDefault();
    api.post('/voitures', form).then(() => navigate('/')); };

  return (
    <form onSubmit={handleSubmit}>
      <h2>Ajouter une voiture</h2>
      <input name="marque" placeholder="Marque" onChange={handleChange} />
      <input name="modele" placeholder="Modèle" onChange={handleChange} />
      <input name="annee" type="number" placeholder="Année"
onChange={handleChange} />
      <input name="couleur" placeholder="Couleur" onChange={handleChange} />
      <button type="submit">Enregistrer</button>
    </form>
  );
}

export default VoitureForm;
```

## 2.6. Composant VoitureEdit.js

```
import React, { useEffect, useState } from 'react';
import api from '../api';
import { useNavigate, useParams } from 'react-router-dom';

function VoitureEdit() {
  const { id } = useParams();
  const navigate = useNavigate();
  const [form, setForm] = useState({
    marque: '',
    modele: '',
    annee: '',
    couleur: '', });

  useEffect(() => {
    api.get(/voitures/${id}).then(res => setForm(res.data));
  }, [id]);

  const handleChange = e => {setForm({ ...form, [e.target.name]: e.target.value });};
};
```

```

const handleSubmit = e => {
  e.preventDefault();
  api.put(/voitures/${id}, form).then(() => navigate('/')); };

return (
  <form onSubmit={handleSubmit}>
    <h2>Modifier une voiture</h2>
    <input name="marque" value={form.marque} onChange={handleChange} />
    <input name="modele" value={form.modele} onChange={handleChange} />
    <input name="annee" type="number" value={form.annee} onChange={handleChange}/>
    <input name="couleur" value={form.couleur} onChange={handleChange} />
    <button type="submit">Mettre à jour</button>
  </form>
);
}

export default VoitureEdit;

```

## 2.7. Mise en place du routing

Dans App.js :

```

import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
import VoitureList from './components/VoitureList';
import VoitureForm from './components/VoitureForm';
import VoitureEdit from './components/VoitureEdit';

function App() {
  return (
    <Router>
      <Routes>
        <Route path="/" element={<VoitureList />} />
        <Route path="/ajouter" element={<VoitureForm />} />
        <Route path="/modifier/:id" element={<VoitureEdit />} />
      </Routes>
    </Router>
  );
}

export default App;

```

- Aller sur <http://localhost:3000> pour afficher la liste des voitures.
- Possibilité de créer, modifier et supprimer une voiture.
- Les données sont stockées dans la base via l'API Laravel.