



7 - Organization du Code

Organiser votre code en modules

Objectif : Structurer votre projet de manière professionnelle.

Plain Text | ...

```
js/ └── config.js // Configuration
      (URL API, constantes) └── api.js //
      Fonctions de communication avec
      l'API └── dom.js // Fonctions de
      manipulation du DOM └── utils.js //
      Fonctions utilitaires (validation,
      formatage) └── home.js // Logique
      spécifique à index.html └── car-
      details.js // Logique spécifique à
      car.html └── script.js // Point
      d'entrée principal
```

Exemple de config.js :

JavaScript | ...

```
export const API_CONFIG = {
  BASE_URL: 'https://votre-
api.onrender.com', ENDPOINTS: {
    CARS: '/cars', CAR_BY_ID: (id) =>
    `/cars/${id}` } }; export const
MESSAGES = { // ... vos messages };
export const VALIDATION = {
  MIN_YEAR: 1900, MAX_YEAR: new
Date().getFullYear() };
```

Exemple de api.js :

JavaScript | ...

```
import { API_CONFIG } from
'./config.js'; export async function
fetchAllCars() { // ... votre code }
export async function
fetchCarById(id) { // ... votre code
} export async function
createCar(carData) { // ... votre
code } export async function
deleteCar(id) { // ... votre code }
```

Exemple de dom.js :

JavaScript | ...

```
export function createCarCard(car) {  
// ... votre code } export function  
displayCars(cars, container) { //  
... votre code } export function  
showLoading(container) { // ...  
votre code } export function  
showError(container, message) { //  
... votre code }
```

Avantages :

- Code organisé et facile à naviguer
- Réutilisabilité des fonctions
- Facilite le travail en équipe
- Simplifie les tests

