

## Tuto

Comment ajouter des pages/fonctionnalités dans le code ???

### **Dossier BACKEND :**

#### **1. Créer le modèle (exemple pour un modèle "Piece")**

*# Créer le fichier dans backend/models/piece.js*

Créer la table dans le bdd ( si celle-ci n'existe pas encore)

#### **2. Créer le contrôleur :**

*# Créer le fichier dans backend/controllers/pieceController.js*

*Il gère les requêtes HTTP et interagit avec le modèle pour manipuler les données.*

#### **3. Créer les routes :**

*# Créer le fichier dans backend/routes/pieceRoutes.js*

*Elles associent les URLs aux méthodes du contrôleur.*

#### **4. Ajouter les routes dans index.js :**

- **Index.js =>** c'est le point d'entrée de l'application. On y importe et configure toutes les routes.

### **Dossier FRONTEND :**

#### **1. Créer le service :**

*# Créer le fichier dans frontend/src/services/pieceService.js*

Le service gère toutes les communications avec l'API backend. Il centralise les appels HTTP.

#### **2. Créer le composant Vue :**

*# Créer le fichier dans frontend/src/views/GestionPieces.vue*

Le composant Vue gère l'interface utilisateur. Il contient le template HTML, la logique JavaScript et les styles CSS.

### 3. Ajouter la route :

*La route définit l'URL qui affichera le composant.*

```
JS javascript ▸ Apply to authService....  
  
// Dans frontend/src/router/index.js  
import GestionPieces from '../views/GestionPieces.vue';  
  
const routes = [  
  {  
    path: '/pieces',  
    name: 'GestionPieces',  
    component: GestionPieces  
  }  
  // Autres routes...  
];
```

### 4. Ajouter le lien dans la navigation :

*Le lien dans la navigation permet aux utilisateurs d'accéder à la page.*

*Dans le frontend/src/App.vue*

***Pour tester :***

1. Backend : **cd backend && node index.js**
2. Frontend : **cd frontend && npm run serve**