

Documentation Technique EasySave

1. Introduction

EasySave est une application de sauvegarde développée en C# suivant le modèle MVC (Modèle-Vue-Contrôleur). Elle permet de créer, lister et exécuter des sauvegardes en utilisant deux stratégies différentes (complète et différentielle). Pour simplifier l'utilisation et le déploiement, l'application gère automatiquement sa configuration minimale et la persistance de ses données.

2. Configuration Minimale

L'application ne requiert aucune configuration externe manuelle avant son exécution. **EasySave** génère automatiquement les fichiers et dossiers nécessaires dans le répertoire d'exécution. **Il est toutefois indispensable que le système dispose des droits d'écriture dans ce répertoire.**

Prérequis Système :

- Une version appropriée de .NET (compatible avec l'application).
- Droits d'écriture sur le répertoire où l'application est exécutée.

Les éléments générés automatiquement sont :

- Un fichier de sauvegarde de la configuration des backups : `backups.json`
- Un fichier d'état des sauvegardes en cours et terminées : `state.json`
- Un dossier de logs : `Logs` (contenant des fichiers journaux au format JSON)

3. Emplacement des Fichiers de Configuration

Tous les fichiers de configuration et de log sont placés dans le **répertoire d'exécution** de l'application (accessible via `EasySave\EasySave\bin\Debug\net9.0`). Voici les détails :

3.1. Fichier `backups.json`

- **Rôle :**
Ce fichier stocke la liste des sauvegardes créées par l'utilisateur. Il contient, pour chaque sauvegarde, des informations telles que :
 - Le nom de la sauvegarde
 - Le chemin source et le chemin cible
 - Le type de sauvegarde (par exemple, "full" pour une sauvegarde complète ou "diff" pour une sauvegarde différentielle)
- **Emplacement :**
`EasySave\EasySave\bin\Debug\net9.0/backups.json`
- **Format :**
JSON, avec une structure hiérarchique permettant la persistance des paramètres de chaque sauvegarde.

3.2. Fichier `state.json`

- **Rôle :**
Ce fichier enregistre l'état et la progression des sauvegardes en cours (ou terminées). Les informations sauvegardées incluent :
 - Le nom de la sauvegarde
 - Le chemin du fichier source et cible en cours de traitement
 - Le statut (NotStarted, Active, End, Error)
 - Le nombre de fichiers à copier et la progression (%)
 - D'autres statistiques relatives à l'exécution
- **Emplacement :**
`EasySave\EasySave\bin\Debug\net9.0/state.json`
- **Format :**
JSON, facilitant le suivi et le contrôle des opérations de sauvegarde.

3.3. Dossier `Logs`

- **Rôle :** Ce dossier contient les fichiers journaux générés lors des opérations de sauvegarde. Chaque opération (copie réussie ou erreur) est consignée avec des informations détaillées telles que :
 - Le nom de la sauvegarde
 - Les chemins des fichiers source et cible
 - La taille du fichier et le temps de transfert
 - La date et l'heure de l'opération
- **Emplacement :**
`EasySave\EasySave\bin\Debug\net9.0\Logs`

- **Format des fichiers :**
Chaque fichier de log porte le nom suivant :
`YYYY-MM-DD-Log.json`
(par exemple, `2025-02-06-Log.json`), et contient des entrées en JSON avec une indentation pour une meilleure lisibilité.

4. Fonctionnement et Processus de Sauvegarde

4.1. Création d'une Sauvegarde

- **Processus :**
 1. L'utilisateur lance l'interface console qui propose une sélection bilingue (Français/Anglais).
 2. L'utilisateur fournit le nom de la sauvegarde, le chemin source (qui doit exister) et le chemin cible (qui peut être créé par l'application si inexistant).
 3. Le type de sauvegarde est spécifié (`full` pour une sauvegarde complète ou `diff` pour une sauvegarde différentielle).
- **Enregistrement :**
Les informations sont sauvegardées dans le fichier `backups.json` et la stratégie correspondante est instanciée automatiquement.

4.2. Exécution d'une Sauvegarde

- **Mise à jour de l'état :**
Lors de l'exécution, l'état de la sauvegarde (progression, fichiers traités, etc.) est mis à jour en temps réel dans le fichier `state.json`.
- **Log des opérations :**
Chaque opération de copie (ou erreur) est consignée dans un fichier du dossier `Logs`.
- **Stratégies de Sauvegarde :**
 - **Sauvegarde Complète (FullBackupStrategy) :** Copie l'ensemble des fichiers depuis le répertoire source vers le répertoire cible.
 - **Sauvegarde Différentielle (DifferentialBackupStrategy) :** Copie uniquement les fichiers nouveaux ou modifiés.

5. Interface Utilisateur et Localisation

- **Interface Console :**
L'application dispose d'une interface utilisateur en mode console qui guide l'utilisateur dans la création et l'exécution des sauvegardes.
- **Localisation (Multilingue) :**
La sélection de la langue (Français ou Anglais) se fait dès le démarrage via l'interface. Les messages affichés sont définis dans des dictionnaires intégrés au code (`Messages.French` et `Messages.English`).

6. Déploiement et Maintenance

- **Déploiement :**
 - Installer l'exécutable et les bibliothèques associées dans un répertoire de votre choix.
 - Veiller à ce que ce répertoire dispose des droits d'écriture, car les fichiers de configuration (`backups.json`, `state.json`) et le dossier `Logs` y seront créés automatiquement.
- **Maintenance :**
 - **Modification de la Configuration :**
Les fichiers de configuration sont mis à jour automatiquement par l'application. Toute modification manuelle doit être réalisée avec précaution.
 - **Consultation des Logs :**
Pour diagnostiquer ou vérifier le déroulement des sauvegardes, consultez les fichiers JSON présents dans le dossier `Logs`.

7. Conclusion

L'application **EasySave** est conçue pour fonctionner avec une configuration minimale, sans nécessiter d'interventions manuelles pour la création ou la gestion des fichiers de configuration.

Tous les fichiers essentiels — le fichier de sauvegarde des configurations (`backups.json`), le fichier d'état (`state.json`), et le dossier de logs (`Logs`) — **sont générés et mis à jour automatiquement dans le répertoire d'exécution de l'application**. Cela simplifie le déploiement et la maintenance, tout en garantissant que l'application dispose des informations nécessaires pour fonctionner correctement.

