VaultFiles

Secure file management library & Web API for .NET 9, with AES encryption, file type filtering, and async fluent API.

VaultFiles

◆ Features

- Save files asynchronously with optional **password-based AES encryption**
- Fluent API for easy integration
- File type filtering (`.jpg` , `.zip` , etc.)
- Max file size limitation
- Overwrite control and unique naming
- Delete files easily
- Fully compatible with **.NET 9**
- Ready-to-use **Web API with Swagger**

◆ Structure

```
VaultFilesSolution/
│
├─ VaultFiles/              <-- Class Library
│     ├─ Interfaces/
│     ├─ Models/
│     ├─ Services/
│     ├─ Exceptions/
│     ├─ Utils/
│     └─ VaultFiles.csproj
│
├─ VaultFiles.Web/          <-- Web API
│     ├─ Controllers/
│     │     └─ FileController.cs
│     ├─ Program.cs
│     ├─ appsettings.json
│     └─ VaultFiles.Web.csproj
│
└─ VaultFiles.sln           <-- Solution
```

◆ Installation

1. Clone the repository:

```
git clone https://github.com/Mr-Tirion/VaultFiles.git
cd VaultFiles
```

1. Build the solution:

```
dotnet build
```

1. Navigate to Web API project and run:

```
cd VaultFiles.Web
dotnet run
```

◆ Usage – Library

Fluent API Example

```csharp
using VaultFiles.Services.Fluent;

var result = await FileBuilder
    .Save("C:\\Temp\\myfile.jpg")
    .ToFolder("C:\\Uploads")
    .WithPassword("1234")
    .AllowExtensions(".jpg", ".zip")
    .MaxSize(10_000_000)
    .Overwrite()
    .Async();

Console.WriteLine($"File saved: {result.FullPath}");
```

Delete File

```csharp
await new FileService().DeleteFileAsync("C:\\Uploads\\myfile.jpg");
```

◆ Usage – Web API

• **Upload File**

```
POST /api/file/upload
Form Data:
```

```
- file: <choose file>
- password: <optional>
```

- **Delete File**

```
DELETE /api/file/delete?fileName=myfile.jpg
```

- Swagger UI: `https://localhost:<port>/swagger`

◆ Contributing

- Open issues and PRs are welcome
- Keep **Uploads/** folder out of GitHub with `.gitignore`

◆ License

MIT License © **Mr-Tirion**