



DLYoutube

SOUS-TITRE DU DOCUMENT

Arnaud lemettre

MTI2025-Console-Youtube

7 Pages

29/02/2024

Version 1.0

Propriétés du document

Auteur	Arnaud lemettre
Version	1.0
Nombre de pages	7
Références	MTI2025-Console-Youtube
Nom fichier	DLYoutube_part1-v1.0.docx

Table des matières

INTRODUCTION	3
CREATION DE L'ARCHITECTURE	4
LE PROJET	4
ARCHITECTURE	4
AJOUT DES FONCTIONNALITES	5
LIEN AVEC YOUTUBE	5
SAUVEGARDE DES FICHIERS	5
LA COUCHE BUSINESS	5
LIGNE DE COMMANDE	6
MODALITE DE RENDU	7

INTRODUCTION

Le but de ce TP est de manipuler l'environnement .NET Core au travers de la création d'une application console permettant de télécharger des vidéos Youtube

Ce travail est à faire individuellement, tout code similaire sur deux personnes sera considéré comme un travail non rendu et non négociable.

Durant cette série de TP les notions suivantes seront abordées :

- EF Core
- XML
- Tests Unitaires

Ce TP devra être réalisé en utilisant dotnet core ou .NET 7

CREATION DE L'ARCHITECTURE

LE PROJET

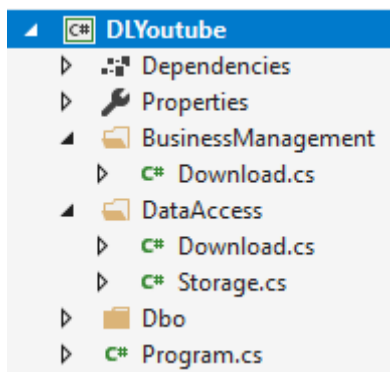
Dans un premier temps vous devez créer un projet Console sous .NET 7

ARCHITECTURE

Vous devrez réaliser une architecture en couche, pour cela vous allez devoir créer 3 dossiers :

- DataAccess
- BusinessManagement
- Dbo

A la fin de ce TP vous devriez avoir cette architecture :



AJOUT DES FONCTIONNALITES

LIEN AVEC YOUTUBE



Afin de pouvoir télécharger les videos, vous allez devoir ajouter un package à votre solution :

<https://www.nuget.org/packages/YoutubeExplode>

Pour manipuler cette bibliothèque, vous allez devoir créer une classe Download dans la DataAccess avec pour méthodes publiques :

- `public async Task<(Stream stream, string title)> DownloadVideo(string urlVideo)`
La méthode prend en paramètre l'url de la video youtube à télécharger, et retourne un tuple avec le stream de la video, ainsi que son titre
- `public async IEnumerable<(Stream stream, string title)> DownloadChannel(string channelId)`
La méthode prend en paramètre l'ID de la chaine Youtube à télécharger. Retourne une liste de tuple avec le stream de la video, ainsi que son titre.

Vous remarquez que nous avons rajouté un nouveau mot clés dans la signature de la méthode : `ASYNc`. Celui-ci va de pair avec le deuxième mot clés : `AWAIT`.

Ces deux mots clés sont du sucre syntaxique pour la gestion de l'asynchronisme en .NET. La majorité des API sous .NET Core sont asynchrones. Nous verrons plus en détails dans les prochains cours comment cela fonctionne. Si vous avez besoin d'informations pour réaliser le TP, vous pouvez regarder ce [contenu](#).

SAUVEGARDE DES FICHIERS

Pour compléter la couche de dataAccess, vous allez devoir créer une classe Storage. Cette classe permettra de sauvegarder les fichiers localement. La classe aura pour fonction :

- `public async Task<bool> SaveFile(Stream stream, string title)`
Stream représente le flux de la video à sauvegarder et title le nom du fichier.

Il faut penser également à gérer les exceptions, ainsi que vérifier le nom du fichier soit compatible avec les règles de nommage des systèmes d'exploitation.

LA COUCHE BUSINESS

Maintenant que nous avons la couche d'accès aux données, nous allons pouvoir rassembler ces fonctionnalités dans la couche de business. Vous allez créer une classe Download.

Cette classe comportera deux méthodes publiques :

- `async Task<bool> DownloadVideo(string[] urlVideos)`
UrlVideos est un tableau contenant l'ensemble des videos à télécharger. Cette méthode appellera la méthode contenue dans la classe Download de la DataAccess pour télécharger la video, puis la méthode SaveFile de la classe Storage. S'il y a une erreur durant le téléchargement la méthode renverra *FALSE* sinon *TRUE*.
- `async Task<bool> DownloadChannel(string channelId)`
ChannelId correspond à l'id de la chaine Youtube à télécharger. S'il y a une erreur durant le téléchargement la méthode renverra *FALSE* sinon *TRUE*.

LIGNE DE COMMANDE

La manipulation de ce programme se fera au travers des commandes suivantes :

```
> .\DLYoutube.exe --help
DLYoutube:
  Permet de telecharger des videos de youtube

Usage:
  DLYoutube [options] [<args> ... ]

Arguments:
  <args>    permet de mettre plusieurs urls de videos à télécharger

Options:
  --channel <channel>  url pour télécharger un channel complet
  --version             Display version information
```

Les différentes options appelleront les méthodes de la couche business.

MODALITE DE RENDU

Les fichiers seront à rendre dans une tarball ayant pour nom :

login_l.zip

Cette tarball devra comprendre à la racine :

- Un dossier contenant la solution Visual Studio qui devra compiler.
Nom : login_IDLYoutube

Attention votre tarball doit être propre, c'est-à-dire ne contenir que des fichiers sources (pas de binaires, de fichiers personnels, ...)

Le tout à envoyer sur l'adresse mti.rendu.dotnet@gmail.com avec les balises suivantes :

[MTI2024][NET][login_l][DLYoutube] partie 1

Où login_l représente votre login.