

Test technique d'intégration dans l'équipe MCO - 2022

Introduction

Bonjour cher collègue !

J'ai essayé de faire un test technique qui se rapproche le plus possible de l'environnement de travail qu'on essaie d'améliorer au quotidien. Voici quelques difficultés que tu pourras rencontrer :

- Peu (voir pas du tout) de documentation
- Les sachants sur nos applicatifs ont quitté la société. Il faut donc tenter de comprendre le fonctionnement (les règles métiers) tout seul en lisant le code.
- Le legacy peut être très lourd. Il y a beaucoup de portions de code difficile à comprendre.

S'il manque des informations, des précisions dans l'énoncé du test, c'est voulu. Ce sont (malheureusement) les conditions dans lesquelles, l'équipe travaille. Bref, bon courage pour le test ! 😊

Prérequis

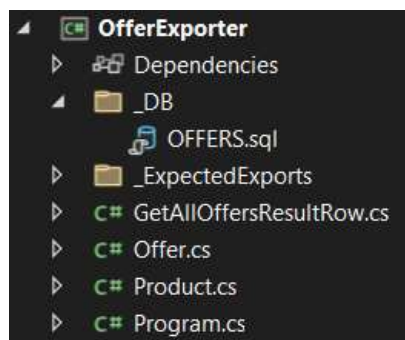
Voici la configuration qui m'a permis de créer le test :

- Solution créée sous Visual Studio 2022 (v. 17.3.3) avec composant pour gérer une base de données (dans le cadre du test, nous utiliserons un fichier .mdf)
- Projet Console App sur NET 6

Avec une version antérieure de Visual Studio ou du framework .NET, le test doit être réalisable mais je n'ai pas essayé. Il est donc préférable d'avoir la même configuration.

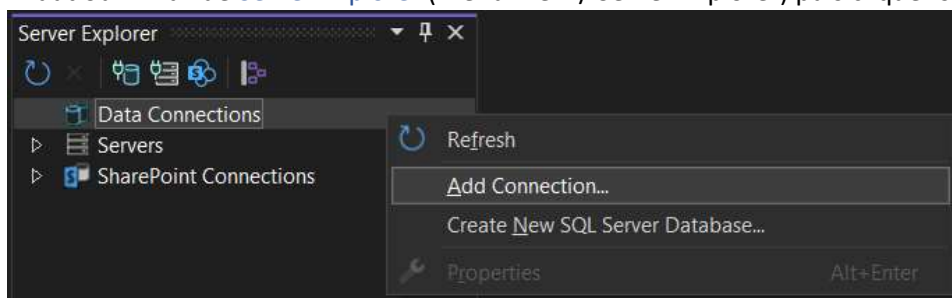
Installation

Dans la solution, il y a un projet [OfferExporter](#) où il manque un fichier [OFFERS.mdf](#)

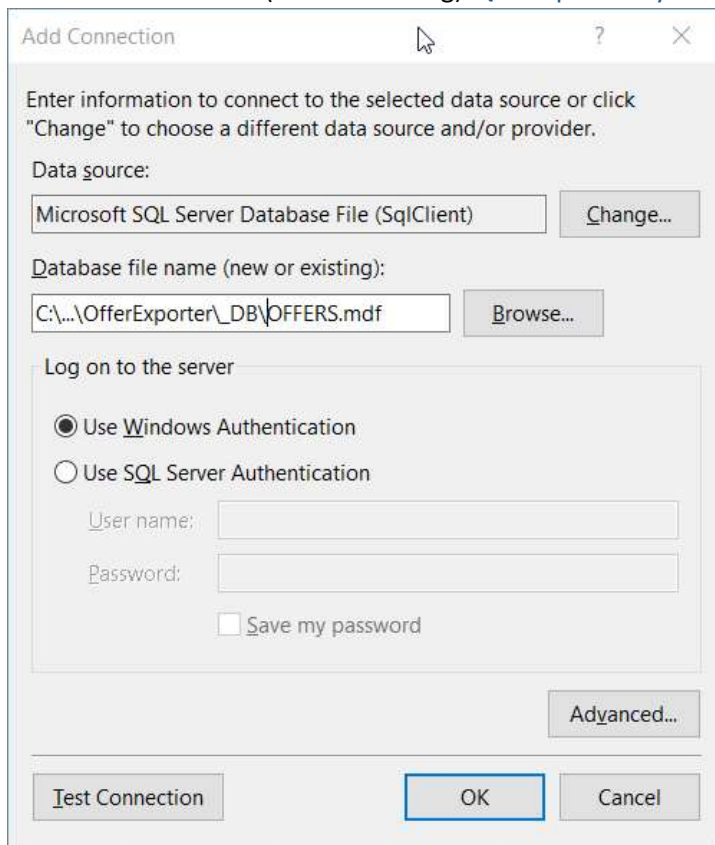


Avant de commencer, il faut créer le fichier [OFFERS.mdf](#). Pour cela, on va le faire manuellement :

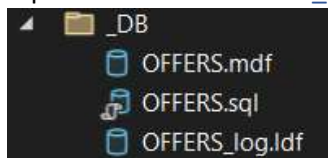
- Il faut ouvrir la vue [Server Explorer](#) (Menu View / Server Explorer) puis cliquer sur [Add Connection...](#)



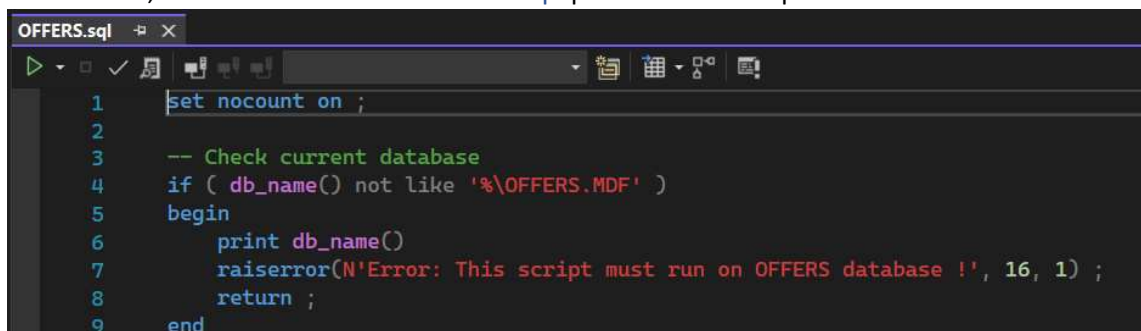
- Dans la fenêtre, renseigner les champs ci-dessous, puis cliquer sur **OK**.
 > Data source : **Microsoft SQL Server Database File (SqlClient)**
 > Database file name (new or existing) : **{local path on your computer}\OfferExporter_DB\OFFERS.mdf**



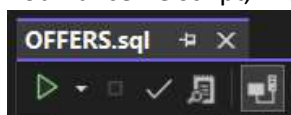
- Une fenêtre de confirmation s'ouvrira, cliquer sur **OK** pour créer le fichier **OFFERS.mdf**
- A présent dans le dossier **_DB**, il devrait y avoir les fichiers créés suivant :



- Maintenant, il faut ouvrir le fichier **OFFERS.sql** qui contient le script d'initialisation de la base :



- Pour lancer le script, il faut se connecter à la base. Pour cela, cliquer sur le bouton de connexion en haut :



- Dans la fenêtre de connexion, renseigner les champs suivant :
 - > Server Name : (localdb)\MSSQLLocalDB
 - > Authentication : Windows Authentication
 - > User Name : {your domain}\{your login}
 - > Database Name : {absolute path to OFFERS.mdf}

Server Name: (localdb)\MSSQLLocalDB

Authentication: Windows Authentication

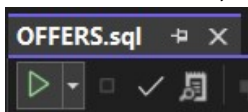
User Name: {your domain}\{your login}

Password:

☐ Remember Password

Database Name: C:\...\OfferExporter\DB\OFFERS.mdf

- Une fois connecté, lancer le script avec le bouton d'exécution en haut à gauche :



- En output, il devrait y avoir les messages suivants :

```
[CREATE TABLES]
12:37:36.006 dbo.Referential created
12:37:36.010 dbo.PromotionTarget created
12:37:36.010 dbo.Seller
12:37:36.010 dbo.Offer created
12:37:36.013 dbo.OfferPromotion created

[INITIALIZE TABLE]
12:37:36.016 dbo.Referential (4 rows inserted)
12:37:36.016 dbo.PromotionTarget (3 rows inserted)
12:37:36.026 dbo.Seller (100 rows inserted)
12:37:48.133 dbo.Offer (100000 rows inserted)
12:37:48.980 dbo.OfferPromotion (10000 rows inserted)

[CREATE STORED PROCEDURES]
12:37:48.986 dbo.GetAllReferentials created
12:37:48.990 dbo.GetAllPromotionTargets created
12:37:48.990 dbo.GetAllSellers created
12:37:48.993 dbo.GetAllOffers created
12:37:48.996 dbo.GetAllOfferPromotions created
```

Cette procédure te sera utile si tu souhaites faire des modifications en base.

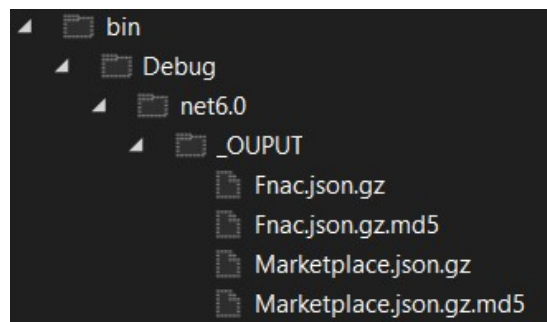
Description du projet

Le projet [OfferExporter](#) est un batch qui exporte pour chaque produit d'un référentiel donné, toutes les offres du catalogue produits. Les exports sont des pseudos-fichiers JSON (une ligne = un JSON d'un produit) zippés au format GZ.

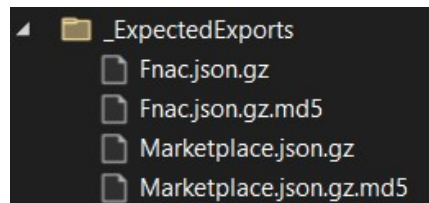
En lançant le projet en mode [Debug](#), on a l'output suivant :

```
14:32:47.670 OfferExport batch started
14:32:48.852 Read offers from database: 100000 found
14:33:05.001 Exclude invalid offers: 62043 left
14:33:51.132 Convert data into products: 61699 products
14:33:51.435 Export Fnac referential: Fnac.json (3522908 bytes)
14:33:51.473 Export Marketplace referential: Marketplace.json (2650960 bytes)
14:33:51.511 Compress Fnac.json: Fnac.json.gz (348584 bytes)
14:33:51.538 Compress Marketplace.json: Marketplace.json.gz (244490 bytes)
14:33:51.542 Compute MD5 Fnac.json.gz: F0A233086D8AF54F8655BDA17CC248D5
14:33:51.544 Compute MD5 Marketplace.json.gz: 376279E7617266A3C2E65FC3894C5C94
14:33:51.545 Clean up working files done
14:33:51.545 OfferExport batch completed in 00:01:03.8751511
```

Les exports sont générés dans un dossier [_OUTPUT](#) où se trouve l'exé :



Dans le projet, il y a un dossier [_ExpectedExports](#) qui contient une version des exports attendus après les évolutions que vous allez apporter à l'application. Vous pourrez comparer les hash MD5 et même dézipper les fichiers pour voir l'attendu.



Evolution à réaliser

C'est parti ! Ci-dessous, voici les tickets à traiter pour le test technique. Tu peux les traiter dans l'ordre que tu préfères.

Ticket 1

- Application : [OfferExporter](#)
- Demandeur : Les consommateurs des exports
- Criticité : [Ultra Critical](#)
- Difficulté : [Low](#)
- **Description** : Actuellement dans les exports, on retrouve des champs [null](#). Par exemple, lorsqu'une offre d'un vendeur n'a pas de promo, on retrouve tout de même le champs [reducedPrice](#) dans le JSON :

```
{ "prid": "1-510", "offers": [ { "seller": "Thome Media", "price": 27.38, "reducedPrice": null, "qty": 1 } ] }
```

Il faudrait faire en sorte que tous les champs [null](#) ne soient plus sérialisés :

```
{ "prid": "1-510", "offers": [ { "seller": "Thome Media", "price": 27.38, "qty": 1 } ] }
```

Ticket 2

- Application : [OfferExporter](#)
- Demandeur : Les consommateurs des exports
- Criticité : [Ultra Critical](#)
- Difficulté : [Medium](#)
- **Description** : Actuellement dans les exports, on retrouve un champ [reducedPrice](#) qui correspond au prix promo d'une l'offre. En revanche, on ne précise pas quelle clientèle cette promotion s'applique. Il faut donc dans l'export de chaque offre, ajouter un champ [discountFor](#) pour indiquer la cible de la promo ([Public](#), [Member](#), [Company \(B2B\)](#))

```
{ "prid": "1-42", "offers": [ { "seller": "VP", "price": 10.00, "reducedPrice": 5.00, "discountFor": "Member", "qty": 2 } ] }
```
- **Commentaire** : Dans l'équipe, personne ne connaît ce batch, ni où se trouve les données demandées.
Bon courage !

Ticket 3

- Application : [OfferExporter](#)
- Demandeur : Tech Lead
- Criticité : [Low](#)
- Difficulté : [Low](#)
- **Description** :
 - Supprimer le [Quick-fix: Exclude Darty referential \(ID 2\)](#)
 - Mettre à jour le champs [IsExportable](#) de Darty dans le fichier [OFFERS.sql](#)
 - Réexécuter le script SQL pour mettre à jour [OFFERS.mdf](#)

Ticket 4

- Application : [OfferExporter](#)
- Demandeur : Tech Lead
- Criticité : [Medium](#)
- Difficulté : [Very High](#)
- **Description** : Le code est lent, consomme trop de mémoire et n'est pas maintenable. Je te laisse refactoriser le code pour améliorer tout ça. Lors du code review, tu me présenteras les améliorations apportées. Il ne faut plus qu'on ait ces [Quick-fix](#) partout...

Modalité de renvoi du test

Pour renvoyer le test, il faudra m'envoyer mail avec la solution zippée en pièce jointe.

Au préalable, pour éviter que la pièce jointe soit bloquée pour des raisons de sécurité ou parce que trop volumineux, il faut :

- Fermer Visual Studio
- Dans le dossier du projet OfferExporter, supprimer les dossiers :
 - bin
 - obj
 - _ExpectedExports
- Dans le dossier _DB, supprimer les fichiers :
 - OFFERS.mdb (je recréerai le fichier à partir du OFFERS.sql)
 - OFFERS_log.ldf
- A la racine de la solution, supprimer :
 - Le dossier caché .vs
 - README.docx
 - README.pdf