Hervé PHILIPPE

ISCA

Résumé

Document pour les développeurs, au sujet du template de génération de sites, AlchiFxCore

Date de création : 17/06/2019

Alchiweb  
Dev – AlchiFxCore

Documentpour l’utilisation d’AlchiFxCore

Table des matières

[Préambule 2](#_Toc11659445)

[AlchiFxCore 2](#_Toc11659446)

[Présentation 2](#_Toc11659447)

[Utilisation 3](#_Toc11659448)

[Installation du modèle de projet 3](#_Toc11659449)

[Création du projet C# pour Visual Studio 3](#_Toc11659450)

[Modifications dans le projet 4](#_Toc11659451)

# Préambule

Ce document a pour but de présenter « AlchiFxCore », outil permettant de générer automatiquement des sites Asp.Net Core avec Entity Framework Core.

# AlchiFxCore

## Présentation

**AlchiFxCore** est un template de projet C# permettant de créer des projets C# pour Visual Studio 2017+ :

* Basé sur **Asp.Net Core** 2.2 (lui-même basé sur .Net Core 2.2)
* Utilisant **Entity Framework Core** 2.2
* Avec des composants web s’inspirant de ceux utilisés dans le [projet opensource Serene](https://github.com/volkanceylan/Serene) (basé sur [Serenity](https://serenity.is/))
* Permettant de générer, à partir des modèles (et donc aussi du contexte de la base de données) :
  + Les *controllers* web Api, gérant le CRUD (pour *create*, *read*, *update*, *delete*)
  + Les *controllers* web gérant le CRUD
  + Les vues (pages Web) liées aux *controllers* web et se servant des *controllers* web Api.

## Utilisation

### Installation du modèle de projet

En ligne de commande (cmd), lancer « **Install-AlchiFxCore** » dans le dossier d’AlchiFxCore :

>Install-AlchiFxCore

Remarques :

* A ne faire qu’une fois
* Pour une mise à jour :
  + Supprimer la version actuelle avec la ligne de commande :

>dotnet new -u Alchiweb.AlchiFxCore

* + Refaire l’installation
* Dans le dossier contenant « **Install-AlchiFxCore.cmd**», il y a aussi le fichier : « **Alchiweb.AlchiFxCore.2.2.1-servicing-t000.nupkg** »
* Dans le fichier « **-AlchiFxCore.cmd** », il y a juste la ligne :

dotnet new -i .\Alchiweb.AlchiFxCore.2.2.1-servicing-t000.nupkgc

### Création du projet C# pour Visual Studio

En ligne de commande (cmd), lancer « **Create-AlchiFxCore** » suivi du nom du projet à créer, dans le dossier d’AlchiFxCore.

Exemples :

>Create-AlchiFxCore Alchi.WebAppKlein

>Create-AlchiFxCore ..\Alchi.WebAppKlein

Remarques :

* Dans le dossier contenant « **Create-AlchiFxCore.cmd**», il y a aussi le fichier : « **Alchiweb.AlchiFxCore.2.2.2.2.1-servicing-t000.nupkg** »
* Dans le fichier « **Create-AlchiFxCore.cmd** », il y a juste la ligne :

dotnet new alchifxcore --auth Individual --use-local-db -o %1

### Modifications dans le projet

#### Copier/générer les modèles

#### Modifier/générer le fichier AlchiDbContext

#### Créer une nouvelle migration

Dans la ***Console du Gestionnaire de package NuGet***, taper :

Add-Migration PremiereMigrationAvecTablesDeBase

#### Mettre à jour la base de données

Par défaut, la base de données est en « **(localdb)**» (cf fichier « ***appsettings.json*** »).

Dans la ***Console du Gestionnaire de package NuGet***, taper :

Update-Database

#### Générer les controllers pour l’Api

Générer automatiquement le *controller* pour l’API web en fonction d’un modèle.

ATTENTION : enlever le « s » ajouté dans le nom du *controller*.

#### Générer les controllers et les vues

Générer automatiquement le *controller* en fonction d’un modèle.

ATTENTION :

* enlever le « s » ajouté dans le nom du *controller*.
* dans chaque *controller* généré, il faut déplacer la portion de code javascript (en haut du fichier du *controller*), dans le fichier « \wwwroot\js\DbTablesDef.js »

Une fois ces étapes faites, les vues https://**[NOM\_DU\_SITE]**/**[NOM\_DU\_MODELE]** sont générées et affichent la table de base du modèle.