

Tutoriel phase 3

But

- Faire une application Web API
- Gérer la persistance des données avec « Entity Framework Core »
- Gérer l'asynchronisme

Développement d'une API en ASP.NET Core Web API pour gérer une bibliothèque

Avant de commencer à faire ce projet il est conseillé d'installer plusieurs extensions de Visual Studio Code :

- C# for Visual Studio Code
- ASP.NET core VS Code Extension Pack

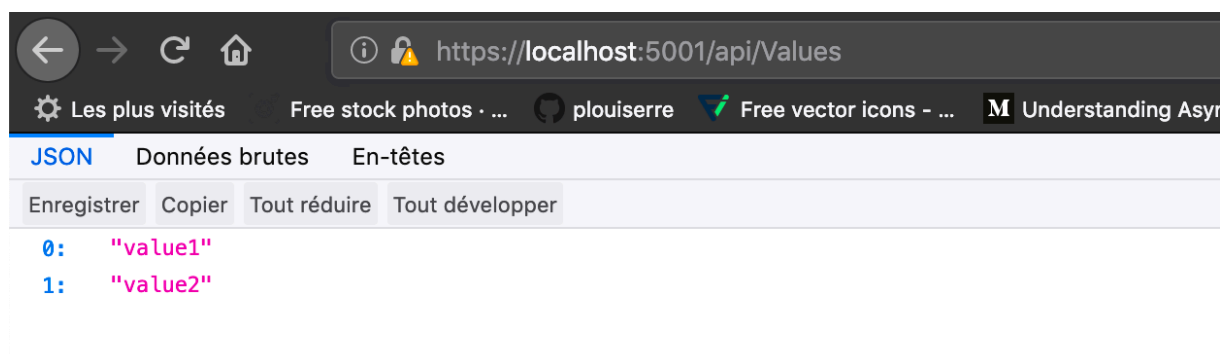
Tout d'abord comme pour le projet précédent il vous faut créer votre projet ASP.NET Core Web API. Pour cela il vous suffit de taper la commande **dotnet new webapi --name *nomDeVotreProjet***.

Votre projet sera ainsi créer avec :

- Program.cs permet de créer l'host pour l'application web
- Startup.cs permet de créer le pipeline de traitement des requêtes de l'application et peut aussi configurer les services de l'application
- DevovxBooksApi.csproj
- Un dossier Controllers avec un Controller « ValuesController.cs » créé par défaut.

Pour voir si votre projet fonctionne correctement il suffit de lancer l'application soit par visual studio code ou soit de taper dans le dossier de votre projet la commande **dotnet run**.

Ensuite à l'aide de votre navigateur aller à l'adresse <https://localhost:5001/api/Values> et vous devriez obtenir cet écran.



Votre API REST aura pour charge de gérer le contenu d'une bibliothèque en exposant 4 méthodes :

- Ajouter un livre dans votre bibliothèque
- Supprimer un livre dans votre bibliothèque
- Retrouver un livre particulier dans votre bibliothèque par son Id
- Retourner tous les livres de la bibliothèque

Pour cela vous devrez utiliser Entity Framework Core avec Sqlite. Entity Framework Core est un ORM en Dotnet Core permettant de gérer avec des objets des bases de données relationnelles. Sqlite est une base de données relationnelles mais ne fonctionne pas avec le système client-serveur. Il s'intègre directement aux programmes. Pour plus de détail n'hésitez pas à consulter ce lien <https://docs.microsoft.com/fr-fr/ef/core/get-started/netcore/new-db-sqlite>

Le modèle de donnée est une classe livre composée de 5 champs :

- Un Id
- Un Titre
- Un Type de livre
- Une année de sortie
- Un nom d'auteur

Ensuite pour pouvoir avoir une API rapide je vous demande d'utiliser l'asynchronisme dans vos API. Pour faire de l'asynchronisme il faut mettre le mot clé async dans la signature de la méthode. De plus quand elle renvoie en mode synchrone un objet X en étant elle renvoie Task<X>. Quand vous appelez une méthode asynchrone dans votre code il faut utiliser le mot clé await avant l'appel de cette méthode. Voici deux liens qui expliquent bien :

- <https://docs.microsoft.com/fr-fr/dotnet/csharp/async>
- <https://docs.microsoft.com/fr-fr/dotnet/standard/async-in-depth>

Pour initialiser le projet :

- Faire un git clone de ce projet github :
 - git clone <https://github.com/plouiserre/DevovxBooksApi.git>
- L'initialisation du projet se trouve sur la branche master

Pour réaliser ce TP, vous aurez 45 minutes pour le réaliser, vous pouvez le tenter seul ou suivre la solution détaillée dans l'autre document.