# ASP .NET Core Web API

## Objectifs

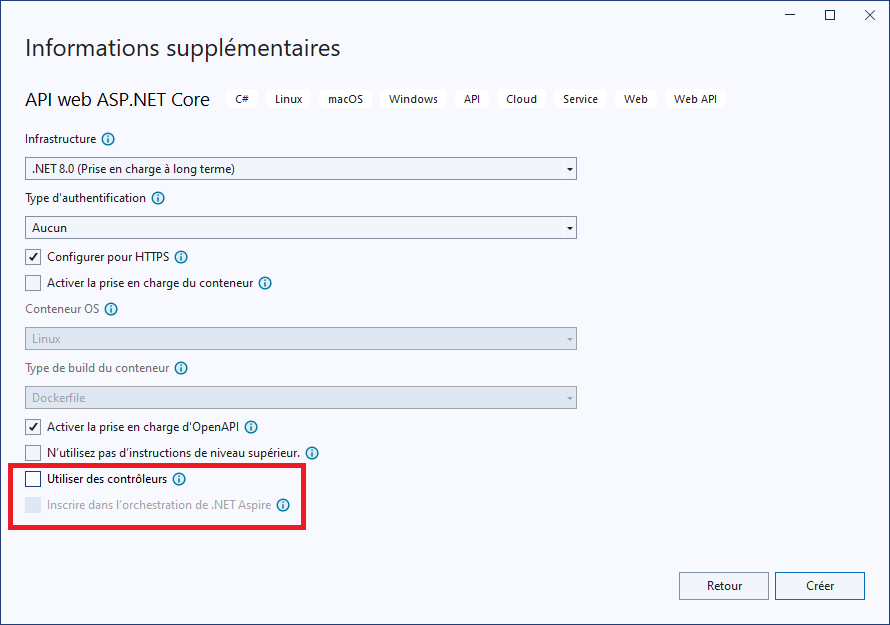
1. Découvrir ASP.NET Core Web API
2. Découvrir Swagger
3. Créer un Web Service minimal et le consommer
4. Créer un Web Service avec Controller et le consommer

## Remarque préliminaire

Ne pas créer son projet sur le disque U si vous êtes sur un PC de l’école. Créez votre projet sur le bureau et à la fin de la séance copiez-le sur le U.

## Exercice 1 – Bank\_Minimal\_API

1. Créer un projet ASP.NET Core Web API que vous nommerez Bank\_Minimal\_API
2. **Décocher la case “use controllers” pour créer un API minimale**



1. Ajouter une route GET pour calculer la tva d’un produit
   1. Cette route prendra 2 paramètres le prix d’un article et le code du pays d’application de la TVA (‘BE’ ou ‘FR’ uniquement)
   2. Exemple du résultat attendu
      1. price : 100, country : ‘BE’ -> 121 (TVA Belgique 21%)
      2. price : 100, country : ‘FR’ -> 120 (TVA France 20%)
   3. Faites au plus simple, le but est simplement de créer une première API !
2. Tester votre API avec Swagger
3. Nettoyer le code pour ne conserver que ce qui est utile

## Exercice 2 – Student\_API\_Controllers

Nous allons maintenant créer une première API avec Controllers pour gérer des étudiants. Cette API sera relativement simple.

L’API contiendra une liste d’étudiants :

private static List<Student> \_students = new List<Student>()

{

new Student { Id = 1, FirstName = "Paul", LastName = "Ochon", Birthdate = new DateTime(1950, 12, 1) },

new Student { Id = 2, FirstName = "Daisy", LastName = "Drathey", Birthdate = new DateTime(1970, 12, 1) },

new Student { Id = 3, FirstName = "Elie", LastName = "Coptaire", Birthdate = new DateTime(1980, 12, 1) }

};

L’API permettra d’afficher la liste des étudiants, d’ajouter un étudiant et de récupérer un étudiant sur base de son Id. **Vous devez définir un modèle “Student” qui sera notamment utilisé pour l’ajout d’un étudiant.**

Vous testerez votre API avec Swagger.

Activer le logging HTTP !