

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

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**

Additional information

ASP.NET Core Web API C# Linux macOS Windows Cloud Service Web

Framework ⓘ
.NET 6.0 (Long-term support)

Authentication type ⓘ
None

☒ Configure for HTTPS ⓘ
☐ Enable Docker ⓘ

Docker OS ⓘ
Linux

☐ Use controllers (unchecked to use minimal APIs) ⓘ
☒ Enable OpenAPI support ⓘ

Back Create

3. Ajouter une route GET pour calculer la tva d'un produit
 - a. Cette route prendra 2 paramètres le prix d'un article et le code du pays d'application de la TVA ('BE' ou 'FR' uniquement)
 - b. Exemple du résultat attendu
 - i. price : 100, country : 'BE' -> 121 (TVA Belgique 21%)
 - ii. price : 100, country : 'FR' -> 120 (TVA France 20%)
 - c. Faites au plus simple, le but est simplement de créer une première API !
4. Tester votre API avec Swagger
5. 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 !