

## ASP.NET Core introductie

### 1 Oefening 1 - Opzet project

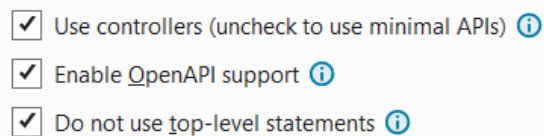
#### LEERDOELEN

1. Je kan het gepaste project type kiezen voor een ASP.NET Core Web API.
2. Je kan het Web API project aanmaken.

#### OEFENING

*Het project dat je in dit labo aanmaakt zal je ook volgende labo(s) nog verder kunnen uitwerken. Het gaat om een API omtrent een winkelketen MyGamestore waarbij niet enkel de winkels, maar ook de werknemers van de winkels zullen kunnen worden beheerd.*

1. Maak vanuit Visual Studio een nieuw ASP.NET Core Web API project aan.
2. Gebruik hierbij de volgende parameters
  - a. Naam project: "MyGameStore"
  - b. Locatie: zelf te kiezen
  - c. Framework versie: .NET 6 (of .NET 7)
  - d. Overige settings op defaults laten staan

- 
- ☒ Use controllers (unchecked to use minimal APIs) ⓘ
  - ☒ Enable OpenAPI support ⓘ
  - ☒ Do not use top-level statements ⓘ

3. Verwijder de Weatherforecast controller en bijhorende class.
4. Start het project op en valideer op die manier of alle NuGet packages correct werden geïnstalleerd.

## 2 Oefening 2 - Eenvoudige controller

### LEERDOELEN

1. Je kan een controller aanmaken voor een Web API.
2. Je kan de nodige Actions implementeren voor een basis CRUD.
3. Je kan de juiste VERB toekennen aan een Action.
4. Je kan een route instellen voor een Action.

### OEFENING

*In deze oefening gaan we de controller voorzien voor de werknemers (People) Met behulp van deze controller kunnen we personen opvragen, toevoegen, aanpassen en verwijderen. Deze controller gaat nog niet werken in combinatie met een database maar gewoon met een lijst van personen in het geheugen van onze API.*

1. Maak in het project een nieuwe lege API controller aan (Geen MVC Controller).
2. Geef deze controller de naam "**PeopleController**".
3. Voeg de nodige **Actions** toe om de basis CRUD operaties te kunnen voorzien.
4. Pas voor elke Action de juiste **VERB** toe en voorzie de nodige **parameters**.
5. Pas waar nodig ook de **Route** aan.

## 3 Oefening 3 - Entiteit klasse van een datamodel

### LEERDOELEN

1. Je kan een entiteit klasse van een datamodel uitwerken.

### OEFENING

*In deze oefening gaan we het datamodel aanmaken dat voorlopig uit slechts 1 entiteit zal bestaan (Person).*

1. Voorzie de klasse **Person**.
2. Geef deze klasse enkele properties (bv. Id, LastName, FirstName, Gender, ...).  
Werk daarbij een getter en setter uit.
3. Pas de constructor van de PeopleController aan zodat hierin een lijst van People wordt voorzien. Je kan deze Lijst al opvullen met de personen die je in het Excel bestand kan terugvinden. Het is daarbij niet de bedoeling om het Excel bestand vanuit code in te lezen maar de namen hardcoded toe te voegen aan de lijst.

## 4 Oefening 4 - Actions implementeren

### LEERDOELEN

1. Je kan een controller aanmaken voor een Web API.
2. Je kan de nodige Actions implementeren voor een basis CRUD.
3. Je kan de juiste VERB toekennen aan een Action.
4. Je kan een route instellen voor een Action.

### OEFENING

*In deze oefening gaan we de Actions implementeren. Als je een action hebt geïmplementeerd kan je deze onmiddellijk testen met behulp van de **Swagger** page. Deze page wordt getoond bij het opstarten van je API. **Test elke action echter ook steeds met behulp van Postman** zodat je deze tool ook leert kennen! Je zal deze nog nodig hebben als je andere online APIs gaat moeten testen. Bij een externe API heb je immers geen swagger page.*

1. Implementeer de action voor het **opvragen** van alle **personen**.
  - a. Bij succesvolle uitvoering moet de action status code 200 (Ok) terugsturen naar de client.
  - b. De body van de response moet een lijst met personen bevatten.  
(deze kan ook leeg zijn in geval je geen personen hebt voorzien in de lijst)
2. Implementeer de action voor het **toevoegen** van een **persoon**.
  - a. De persoon die door de client wordt verstuurd moeten we dus toevoegen aan de lijst in de controller.
  - b. Je voorziet hierbij zelf de Id voor deze persoon. Merk op dat dit normaal gezien door de database wordt voorzien bij nieuwe personen.
  - c. Bij succesvolle toevoeging moet de action status code 201 (Created) terugsturen naar de client.
  - d. De body van de response bevat de toegevoegde persoon met de Id ingevuld.

*Merk op dat als je nu eerst een persoon toevoegt via een **POST** request en onmiddellijk nadien terug via een **GET** request alle personen opvraagt, dat je de nieuwe persoon **niet zal terugvinden** in de response....  
Denk na wat de oorzaak hiervan zou kunnen zijn en tracht hiervoor een oplossing te bedenken.*

3. Implementeer de action voor **verwijderen** van een **persoon**.
  - a. Zoek de persoon op in de lijst en verwijder deze.
  - b. Bij succesvolle verwijdering moet de action status code 204 (NoContent) terugsturen naar de client.
  - c. Indien de persoon niet in de lijst voorkomt wordt een status code 404 gegeven. Zoek zelf op welk ActionResult hiervoor gebruikt moet worden.

4. Implementeer de action voor het **aanpassen** van een **persoon**.
  - a. Zoek de persoon op.
  - b. Pas de properties aan.
  - c. Bij succesvolle aanpassing moet de action status code 200 terugsturen (Ok) naar de client
  - d. De body van de response bevat de aangepaste persoon.
  - e. Indien de persoon niet in de lijst voorkomt wordt een status code 404 gegeven.
5. Voorzie en implementeer een extra Action voor het **opvragen** van een persoon **aan de hand van de Id**.
  - a. Denk zelf na wat er nodig is om deze action te implementeren, wat er kan mislopen en welke status code terug te geven.
6. Voorzie en implementeer een extra Action voor het opvragen van alle **personen met een bepaalde LastName**.
  - a. De LastName waarop moet gezocht worden moet als query parameter meegestuurd worden vanuit de client.
  - b. De body van de response bevat de lijst met personen waarvan de LastName overeenkomt met de waarde van de bijhorende query parameter.
7. Pas de Action voor het verwijderen van een persoon aan zodat enkel verwijderd mag worden als de juiste "key" wordt meegegeven.
  - a. De "key" wordt meegestuurd via het **header** veld "**X-AccessKey**".
  - b. De **waarde** van deze key moet ingesteld staan op "**123456789**".
  - c. Wanneer geen geldige key wordt meegegeven moet de Action het ActionResult "**Unauthorized**" teruggeven.