

Eenvoudige Web API opzetten

Oefening 1: Project voorbereiden

- Maak vanuit Visual Studio een nieuw **ASP.NET Core Web API** project.
- Geef het project de naam 'MyGameStore' en kies de locatie van het project.
- Gebruik .NET 5 als framework versie.
- Bij de overige settings mag je de 'default' waarde behouden.
- Start het project even op om te testen of de aanmaak en installatie van de nuGet packages correct is verlopen.
- De Weatherforecast controller en bijhorende WeatherForecast class mag je eventueel verwijderen.

Oefening 2: Controller aanmaken

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 Gamestore waarbij niet enkel de winkels, maar ook de werknemers van de winkels zullen kunnen worden beheerd. In dit labo gaan we enkel de controller voorzien voor de werknemers (People).

- Als eerste gaan we dus een **PeopleController** voorzien waarmee we Personen kunnen opvragen, aanpassen, toevoegen en verwijderen. Voeg een nieuwe lege **API** controller (dus geen MVC Controller) toe aan je project.
- We voorzien dus de nodige **Actions** op de controller om dit te verwezenlijken.
- Bij elke action gebruik je de juiste **VERB** en voorzie je de nodige **parameters**.
- Pas waar nodig ook de **Route** aan.
- Deze controller gaat nog niet werken in combinatie met een databank, maar gewoon met een lijst van Personen in het geheugen van onze API.

Oefening 3: Model aanmaken

- Ons datamodel zal enkel bestaan uit 1 klasse: **Person**.
- Voorzie deze klasse en voeg enkele properties toe (bv. Id, LastName, FirstName, Gender, ...) met een getter/setter.
- Vervolgens gaan we een Lijst van personen voorzien in onze controller, maak deze aan in de constructor.
- Vul de lijst van personen in de controller constructor tevens op met bv. 2 fictieve personen zodat je reeds over test gegevens beschikt als je nadien via de API personen gaat opvragen.

Oefening 4: Actions implementeren

*Als je een action hebt geïmplementeerd kan je deze uiteraard onmiddellijk testen met behulp van de **Swagger** page die wordt getoond bij het opstarten van je API. **Test echter ook steeds met behulp van Postman elke action!** Hiermee leer je tevens Postman kennen, wat je zal nodig hebben als je andere online APIs gaat moeten testen (want bij een externe API heb je uiteraard geen swagger page).*

- Implementeer de action voor het **opvragen** van alle **personen**.
 - Als het gelukt is moet de API een status code 200 terugsturen naar de client met in de body een lijst met personen (kan ook een lege lijst zijn eventueel als er geen personen zijn).
- Implementeer de action voor het **toevoegen** van een **persoon**.
 - De persoon die door de client wordt doorgestuurd moeten we dus toevoegen aan de lijst in de controller.
 - Merk op dat normaal gezien de databank automatisch een Id zal toekennen aan nieuwe personen. Dit zal je dus nu ook zelf moeten voorzien in de controller bij het toevoegen.
 - Als het gelukt is wordt een status code 201 (created) teruggestuurd naar de client, in de body zal de persoon worden teruggestuurd met de nieuwe 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.*

- Implementeer de action voor het **verwijderen** van een persoon.
 - Zoek de persoon op in de lijst en verwijder deze.
 - Als het gelukt is wordt een status code 204 (NoContent) teruggestuurd naar de client.
 - Indien de persoon niet in de lijst voorkomt wordt een status code 404 gegeven.
- Implementeer de action voor het **aanpassen** van een persoon.
 - Zoek de persoon op.
 - Pas de properties aan.
 - Als het gelukt is wordt een status code 200 teruggestuurd met in de body de aangepaste persoon.
 - Indien de persoon niet in de lijst werd teruggevonden wordt een status code 404 gegeven.

- Voorzie en implementeer een extra Action voor het **opvragen** van een persoon **aan de hand van de Id**.
 - Denk zelf na wat er nodig is om deze action te implementeren, wat er kan mislopen en geef de juiste status code terug.
- Voorzie dat het mogelijk wordt om aan de API niet enkel alle personen op te vragen, maar ook enkel **personen met een bepaalde LastName**.
 - De LastName waarop moet gezocht worden wordt dan als query parameter meegestuurd vanuit de client.
 - Overloop de lijst van personen en filter de perso(o)n(en) met de betreffende naam eruit, stuur deze **lijst** vervolgens terug in de respons.
- Als laatste voorzie je dat verwijderen van een persoon enkel kan met behulp van de juiste **"key"**.
 - De "key" wordt meegestuurd via het **header** veld **"X-AccessKey"**.
 - De **waarde** van deze key moet ingesteld staan op **"123456789"**.
 - Zonder deze key in de header zal een delete niet toegelaten zijn en geef je de status code **Unauthorized** terug.