2022

Thijmen Hogenkamp & Stefan Verkuijlen

O2GO

3-11-2022

Testrapport Back-End

Afbeelding met zwart

Automatisch gegenereerde beschrijving

# Inleiding

In dit document worden een paar uitgevoerde testen beschreven. Hierin is het belangrijkste wat de functionaliteit moet doen, de uitgevoerde stappen, het verwachte resultaat en het gegeven resultaat.

# Requirements

Hieronder staan de requirements die opgesteld waar wij de testen van gaan laten zien.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Beschrijving** | **MoSCoW** | **Type** |
| **R01** | De gebruiker moet kunnen registreren | Must | Functioneel |
| **R02** | De gebruiker moet kunnen inloggen | Must | Functioneel |
| **R08** | Als admin moet ik scooters kunnen aanmaken, aanpassen en verwijderen | Must | Functioneel |

# Uitgevoerde testen

Hier worden uitgevoerde testen beschreven. Deze zullen onderverdeeld worden in bad- en good cases. We gaan ervan uit dat alle calls gedaan worden middels een call-testing programma genaamd Postman.

## Geautomatiseerde testen

De geautomatiseerde testen die geschreven zijn kunnen worden uitgevoerd door middel van een command in de terminal. In de root (O2GOBackEnd) kun je de command uitvoeren: *dotnet test*. Deze command zorgt dat alle testen uitgevoerd worden en geven hierop een resultaat. Het resultaat zal er dan als volgt uit moeten zien:

Afbeelding met tekst

Automatisch gegenereerde beschrijving

## Handmatige testen

Hier worden de handmatig uitgevoerde testen beschreven.

### Good case

|  |  |
| --- | --- |
| **Titel: Scooter aanmaken** | **R08** |
| *Test:*  Een admin moet een scooter kunnen aanmaken. | |
| *Uitgevoerde stappen:*   1. De API starten 2. Call wordt op POST gezet 3. Link wordt ingevuld (<https://localhost:7283/api/scooters/create> ) 4. Body wordt gevuld   {  "licensePlate": "DFJ85X",  "maxKmh": 30,  "brand": "AGM",  "year": 2020,  "description": "Mooie scooter, 5500 op de teller. Altijd netjes onderhouden.",  "price": 60.30  }   1. De call wordt verstuurd | |
| *Verwacht resultaat:*  De scooter wordt toegevoegd en het object wordt netjes teruggezonden. | |
| *Gegeven resultaat:*  {  "licensePlate": "DFJ85X",  "maxKmh": 30,  "brand": "AGM",  "year": 2020,  "description": "Mooie scooter, 5500 op de teller. Altijd netjes onderhouden.",  "price": 60.30  } | |

### Bad case

|  |  |
| --- | --- |
| **Titel: Inloggen gebruiker** | **R02** |
| *Test:*  Een gebruiker moet met een email en wachtwoord kunnen inloggen. | |
| *Uitgevoerde stappen:*   1. De API starten 2. Call wordt op POST gezet 3. Link wordt ingevuld (<https://localhost:7283/api/auth/login>) 4. Body wordt gevuld   {  "Email": "test@admin.com",  "Password": "Test123!blablabla",  }   1. De call wordt verstuurd | |
| *Verwacht resultaat:*  Dat er een fout gegeven wordt met status 401 unauthorized | |
| *Gegeven resultaat:*  *401 Unauthorized* | |

|  |  |
| --- | --- |
| **Titel: Registreren gebruiker** | **R01** |
| *Test:*  Een gebruiker moet met een email, wachtwoord, voornaam, achternaam en adresgegevens kunnen registreren. | |
| *Uitgevoerde stappen:*   1. De API starten 2. Call wordt op POST gezet 3. Link wordt ingevuld (<https://localhost:7283/api/auth/register>) 4. Body wordt gevuld   {  "Email": "test@admin.com",  "Password": "Test123!",  "FirstName": "Jan",  "LastName": "Janssen",  "Street": "Janssenlaan",  "Housenumber": 43,  "PostalCode": "7543GJ",  "City": "Enschede"  }   1. De call wordt verstuurd | |
| *Verwacht resultaat:*  Er wordt verwacht dat de API een nette foutmelding teruggeeft wat laat zien dat het e-mailadres al bestaat. | |
| *Gegeven resultaat:*  Email stefan5@admin.com already taken. | |