Back-end: Spring Boot: D&D Character Generato.

Thomas Gielen.

Scriptie voorgedragen tot het bekomen van de graad van
Graduaat in het Programmeren

Promotor: Luc Vervoort
Co-promotor: David Breckx
Academiejaar: 2024–2025
Eerste examenperiode

Departement IT en Digitale Innovatie .

HO
GENT

Woord vooraf

Voor u ligt mijn graduaatsproef, opgesteld in het kader van het behalen van het diploma Graduaat Programmeren.
Voor dit eindproject koos ik ervoor om een RESTful API te ontwikkelen voor het beheren van Dungeons Dragons-personages. Dit onderwerp combineert mijn interesse in games met mijn passie voor backendontwikkeling, en gaf me de kans om een praktische en boeiende toepassing uit te werken.
Een extra uitdaging was het gebruik van Spring Boot, een technologie waarmee ik tot dit project nog niet had gewerkt. Het was een leerzaam proces waarbij ik veel nieuwe concepten heb ontdekt, en waarbij ik stap voor stap mijn weg heb gevonden in het bouwen van een gestructureerde en functionele API.
Ik wil graag mijn familie en vrienden bedanken voor hun geduld, begrip en steun tijdens dit traject. Hun aanmoediging heeft me geholpen om gemotiveerd te blijven en het project tot een goed einde te brengen.
Met trots presenteer ik het resultaat van mijn inspanningen en hoop ik dat dit project mijn groei als ontwikkelaar weerspiegelt.

Samenvatting

Deze graduaatsproef behandelt de ontwikkeling van een backend API voor het creëren en beheren van Dungeons & Dragons-personages, met behulp van het Java-framework Spring Boot. Het onderwerp is gekozen vanwege de groeiende populariteit van digitale hulpmiddelen in rollenspellen en de behoefte aan flexibele, uitbreidbare oplossingen voor spelers en ontwikkelaars.
Het onderzoek richt zich op de vraag hoe een API ontworpen en geïmplementeerd kan worden die verschillende karakterklassen, rassen en statistieken ondersteunt, terwijl ook gebruikersauthenticatie is geïntegreerd. De doelstelling was het ontwikkelen van een functioneel prototype dat de kernfunctionaliteiten voor het beheren van personages omvat.
De methodologie bestond uit een documentation-first aanpak, waarbij met behulp van Apidog een OpenAPI-specificatie werd opgesteld voorafgaand aan de implementatie. Dit zorgde voor duidelijke richtlijnen tijdens de ontwikkeling en maakte de API goed testbaar en uitbreidbaar.
De resultaten tonen aan dat met Spring Boot een efficiënte en goed gestructureerde API kan worden gerealiseerd die voldoet aan de eisen van de onderzoeksvraag. Het prototype biedt een stabiele basis voor verdere uitbreiding, zoals het toevoegen van een gebruikersinterface of extra spelmechanismen.

Inhoudsopgave

Lijst van figuren

Lijst van tabellen

Lijst van codefragmenten

1

Inleiding

Deze graduaatsproef richt zich op het ontwikkelen van een RESTful API met Spring Boot voor het aanmaken en beheren van Dungeons & Dragons-personages. Dungeons & Dragons (D&D) is een populair rollenspel waarbij spelers unieke personages creëren met eigen rassen, klassen en eigenschappen. De API ondersteunt dit proces digitaal en gestructureerd.
Het project maakt gebruik van Spring Boot, een modern Java-framework dat binnen deze toepassing voor het eerst wordt gebruikt. De API is ontworpen volgens het documentation-first-principe met behulp van Apidog, wat zorgt voor duidelijke en consistente documentatie van de verschillende functionaliteiten.
De toepassing biedt ondersteuning voor het aanmaken, bewerken en verwijderen van personages, inclusief rassen, klassen en gebruikersauthenticatie. Geavanceerde spelmechanismen zoals gevechten of campagnes vallen buiten de scope.
De centrale onderzoeksvraag luidt: Hoe kan met behulp van Spring Boot een goed gestructureerde en bruikbare API ontwikkeld worden voor het beheren van D&D-personages?
Het doel is een functionele en uitbreidbare backendtoepassing te realiseren die voldoet aan moderne ontwikkelingsprincipes.

1.1. Probleemstelling

Het beheren van personages in Dungeons & Dragons gebeurt vaak handmatig of via tools die beperkt, betalend of moeilijk uitbreidbaar zijn. Veel spelers en spelbegeleiders (Dungeon Masters) gebruiken losse documenten, spreadsheets of niet-gecentraliseerde websites, wat leidt tot fouten, onduidelijkheden en verlies van gegevens.
Voor kleine spelgroepen, individuele D&D-spelers en ontwikkelaars van digitale DD-tools is er behoefte aan een eenvoudige, uitbreidbare en gratis backendoplossing

waarmee personages digitaal kunnen worden aangemaakt, bewerkt en bewaard. Vooral gebruikers die een eigen tool, app of interface willen bouwen, hebben nood aan een goed gestructureerde, open API als basis.
Door een gebruiksvriendelijke en goed gedocumenteerde REST API te voorzien, kan deze toepassing een duidelijke meerwaarde bieden aan deze specifieke doelgroep, die vandaag vaak afhankelijk is van gefragmenteerde of commerciële oplossingen.

1.2. Onderzoeksvraag

Hoe kan met behulp van Spring Boot een uitbreidbare en goed gedocumenteerde API ontwikkeld worden die het aanmaken en beheren van Dungeons & Dragons-personages ondersteunt voor kleine spelgroepen en individuele spelers?

1.3. Onderzoeksdoelstelling

Het doel van deze graduaatsproef is het ontwikkelen van een functionele en uitbreidbare RESTful API met behulp van Spring Boot, waarmee Dungeons Dragons-personages kunnen worden aangemaakt, beheerd en verwijderd. De API moet logisch en intuïtief opgebouwd zijn, zodat gebruikers zoals spelers en ontwikkelaars er gemakkelijk mee kunnen werken. Daarnaast is het belangrijk dat de structuur modulair en flexibel is, zodat in de toekomst extra spelregels en functionaliteiten eenvoudig kunnen worden toegevoegd. Om de persoonlijke gegevens van gebruikers te beschermen, wordt er een werkende gebruikersauthenticatie geïmplementeerd. De API zal volledig en duidelijk gedocumenteerd worden via Apidog volgens het documentation-first principe, waardoor externe ontwikkelaars de API eenvoudig kunnen gebruiken en integreren. Dit project wordt gerealiseerd als een proof of concept dat de basisfunctionaliteiten operationeel toont en kan dienen als uitgangspunt voor verdere ontwikkeling of integratie met frontend-applicaties. Met deze graduaatsproef wordt aangetoond dat het mogelijk is om met Spring Boot een degelijke backend voor Dungeons Dragons-personagebeheer te realiseren die aansluit bij de behoeften van kleine spelgroepen en individuele spelers.

1.4. Opzet van deze graduaatsproef

De rest van deze graduaatsproef is als volgt opgebouwd:
In Hoofdstuk 2 wordt een overzicht gegeven van de stand van zaken binnen het onderzoeksdomein, op basis van een literatuurstudie.
In Hoofdstuk 3 wordt de methodologie toegelicht en worden de gebruikte onderzoekstechnieken besproken om een antwoord te kunnen formuleren op de onderzoeksvragen.
In Hoofdstuk 5, tenslotte, wordt de conclusie gegeven en een antwoord geformuleerd op de onderzoeksvragen. Daarbij wordt ook een aanzet gegeven voor toe-

komstig onderzoek binnen dit domein.

2

Stand van zaken

2.1. Stand van Zaken

Dit hoofdstuk geeft een overzicht van de huidige stand van zaken rond digitale hulpmiddelen voor D&D Beyond (**Bradford2025**). De inhoud bouwt voort op de inleiding en spitst zich toe op de technologische ontwikkelingen die het spelen van D&D Beyond digitaal ondersteunen. Zo wordt de lezer volledig geïnformeerd over de state-of-the-art op dit gebied, zodat die het verdere verhaal zonder aanvullende voorkennis kan volgen.

Digitale platforms zoals **D&D Beyond** bieden uitgebreide functionaliteiten voor het creëren en beheren van personages, het bijhouden van campagnes en het integreren van digitale dobbelstenen en kaarten (**Bradford2025**). Andere tools zoals **Roll20** (**Melzer2024**) en **Fantasy Grounds** stellen spelers en Dungeon Masters in staat virtuele tafelopstellingen te gebruiken met geavanceerde functies, waaronder dynamische kaarten, automatische dobbelsteenrollen en geïntegreerde chatfunctionaliteit (**Hall2015**). Hoewel deze platforms een robuuste gebruikerservaring bieden, zijn ze vaak gesloten systemen met beperkte mogelijkheden tot aanpassing en integratie, wat de vraag naar open en uitbreidbare oplossingen versterkt.

Voor de ontwikkeling van webservices wordt vaak gekozen voor **RESTful API's**, die stateless communicatie tussen client en server mogelijk maken en zo schaalbaarheid en eenvoud in het gebruik bevorderen (**restfull**). In de Java-omgeving is **Spring Boot** een gangbaar framework voor het snel opzetten van dergelijke API's, doordat het tal van ingebouwde configuraties en ontwikkeltools aanbiedt (**Pratik2024**). Hierdoor kunnen ontwikkelaars vlot een robuuste backend creëren die gemakkelijk integreert met verschillende frontend-applicaties en systemen.

Een moderne methodologie in API-ontwikkeling is de *API-Design First* benadering, waarbij het ontwerp van de API voorafgaat aan de implementatie. Deze werkwijze bevordert duidelijkheid en verbetert de samenwerking tussen ontwikkelaars

en andere betrokkenen (**apidog**). Het platform **Apidog** ondersteunt deze aanpak door hulpmiddelen te bieden voor visueel ontwerpen, testen en documenteren van API's, wat bijdraagt aan snellere ontwikkeling en hogere kwaliteit.
Tot slot is beveiliging een essentieel aandachtspunt bij API-ontwikkeling, vooral bij het verwerken van gevoelige data. Een veelgebruikte authenticatiemethode is het gebruik van **JSON Web Tokens (JWT)**, die gebruikers op een veilige en efficiënte wijze authenticeren en autoriseren zonder dat servers sessie-informatie hoeven op te slaan (**Gordadze**).

3

Methodologie

3.1. Ontwerp met Apidog

Voor de ontwikkeling van de API werd gekozen voor een *API-Design First* benadering. Hierbij werd direct begonnen met het ontwerpen van de API in **Apidog**, een tool die het mogelijk maakt om API-specificaties visueel en overzichtelijk op te stellen. Met Apidog werden alle benodigde endpoints, request- en responseformaten en validatieregels vastgelegd voordat de daadwerkelijke implementatie startte. Dit zorgde voor duidelijkheid over wat de API moest doen en maakte het mogelijk om vroegtijdig te testen en de documentatie automatisch te genereren. Door deze gestructureerde aanpak werden fouten tijdens de implementatie beperkt en kon het ontwikkelproces efficiënter verlopen.

3.2. Implementatie met Spring Boot

Op basis van het ontwerp uit Apidog werd de backend van de API geïmplementeerd met het **Spring Boot** framework. Spring Boot biedt een snelle en gestroomlijnde manier om RESTful API's te ontwikkelen in Java, met ingebouwde ondersteuning voor onder andere routing, datahandling en beveiliging. Tijdens de implementatie werden functionaliteiten ontwikkeld om Dungeons & Dragons-personages aan te maken, te beheren en gebruikersauthenticatie te ondersteunen. De structuur en specificaties vanuit Apidog dienden als leidraad, zodat de implementatie nauw aansloot bij het vooraf opgestelde ontwerp. Daarnaast werden testen uitgevoerd om te garanderen dat de API correct functioneerde en voldeed aan de gestelde eisen.

4

Ontwerp met Apidog

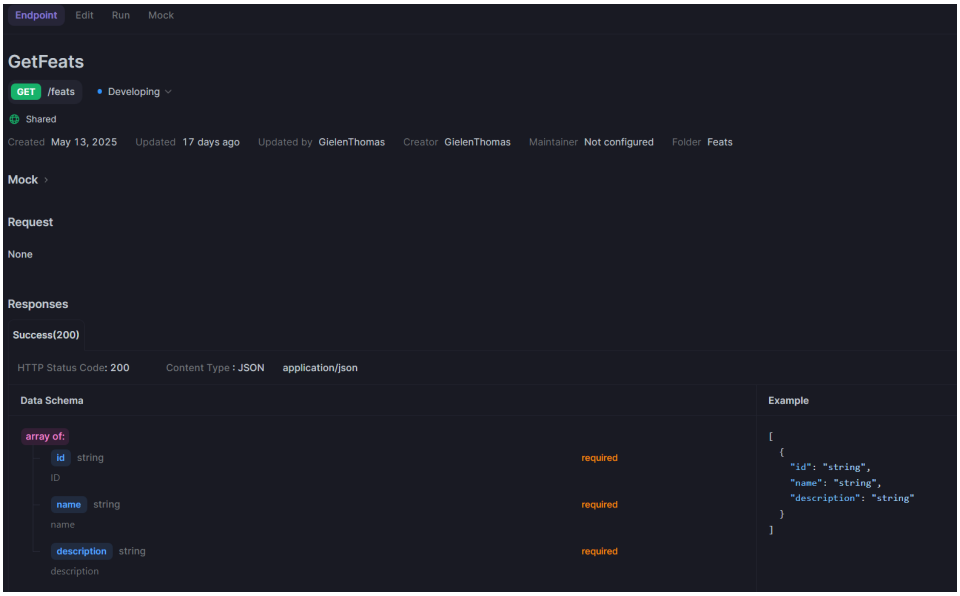
In dit hoofdstuk wordt het gebruik van **Apidog** toegelicht voor het ontwerpen van de API. Apidog maakt het mogelijk om OpenAPI-specificaties visueel op te stellen, te valideren en automatisch te documenteren, wat het ontwikkelproces gestructureerd en transparant maakt.
Met Apidog werden alle API-endpoints, datamodellen en parameters opgesteld volgens de OpenAPI-standaard. Hieronder is een voorbeeld van een datamodel in dogApi. Hieronder staan er voorbeelden van hoe het in de dogAPi er uitziet en wat het genereert.

```
        "/feats": {
            "get": {
                "summary": "GetFeats",
                "deprecated": false,
                "description": "",
                "tags": [],
                "parameters": [],
                "responses": {
                    "200": {
                        "description": "",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "array",
                                    "items": {
                                        "$ref": "#/compo-
                                            nents/schemas/FeatResponse"
                                    }
                                }
                            },
                            "headers": {}
                        }
                    },
                    "security": []
            },
```

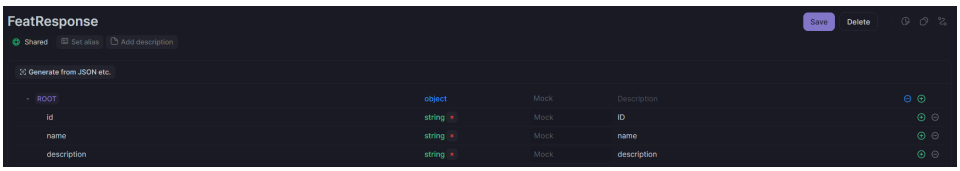
Codefragment 4.1: End-point van de OpenAPI spec

```
        "FeatResponse": {
            "type": "object",
            "properties": {
                "id": {
                    "type": "string",
                    "description": "ID"
                },
                "name": {
                    "type": "string",
                    "description": "name"
                },
                "description": {
                    "type": "string",
                    "description": "description"
                }
            },
            "required": [
            "name",
            "description",
            "id"
            ]
        },
```

Codefragment 4.2: Data model van de OpenApi spec



Figuur 4.1: Voorbeeld van een endpoint in dogApi



Figuur 4.2: Voorbeeld van een data model in dogApi

5

Conclusie

Curabitur nunc magna, posuere eget, venenatis eu, vehicula ac, velit. Aenean ornare, massa a accumsan pulvinar, quam lorem laoreet purus, eu sodales magna risus molestie lorem. Nunc erat velit, hendrerit quis, malesuada ut, aliquam vitae, wisi. Sed posuere. Suspendisse ipsum arcu, scelerisque nec, aliquam eu, molestie tincidunt, justo. Phasellus iaculis. Sed posuere lorem non ipsum. Pellentesque dapibus. Suspendisse quam libero, laoreet a, tincidunt eget, consequat at, est. Nullam ut lectus non enim consequat facilisis. Mauris leo. Quisque pede ligula, auctor vel, pellentesque vel, posuere id, turpis. Cras ipsum sem, cursus et, facilisis ut, tempus euismod, quam. Suspendisse tristique dolor eu orci. Mauris mattis. Aenean semper. Vivamus tortor magna, facilisis id, varius mattis, hendrerit in, justo. Integer purus.

Vivamus adipiscing. Curabitur imperdiet tempus turpis. Vivamus sapien dolor, congue venenatis, euismod eget, porta rhoncus, magna. Proin condimentum pretium enim. Fusce fringilla, libero et venenatis facilisis, eros enim cursus arcu, vitae facilisis odio augue vitae orci. Aliquam varius nibh ut odio. Sed condimentum condimentum nunc. Pellentesque eget massa. Pellentesque quis mauris. Donec ut ligula ac pede pulvinar lobortis. Pellentesque euismod. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent elit. Ut laoreet ornare est. Phasellus gravida vulputate nulla. Donec sit amet arcu ut sem tempor malesuada. Praesent hendrerit augue in urna. Proin enim ante, ornare vel, consequat ut, blandit in, justo. Donec felis elit, dignissim sed, sagittis ut, ullamcorper a, nulla. Aenean pharetra vulputate odio.

Quisque enim. Proin velit neque, tristique eu, eleifend eget, vestibulum nec, lacus. Vivamus odio. Duis odio urna, vehicula in, elementum aliquam, aliquet laoreet, tellus. Sed velit. Sed vel mi ac elit aliquet interdum. Etiam sapien neque, convallis et, aliquet vel, auctor non, arcu. Aliquam suscipit aliquam lectus. Proin tincidunt magna sed wisi. Integer blandit lacus ut lorem. Sed luctus justo sed enim.

Morbi malesuada hendrerit dui. Nunc mauris leo, dapibus sit amet, vestibulum et, commodo id, est. Pellentesque purus. Pellentesque tristique, nunc ac pulvinar adipiscing, justo eros consequat lectus, sit amet posuere lectus neque vel augue. Cras consectetuer libero ac eros. Ut eget massa. Fusce sit amet enim eleifend sem dictum auctor. In eget risus luctus wisi convallis pulvinar. Vivamus sapien risus, tempor in, viverra in, aliquet pellentesque, eros. Aliquam euismod libero a sem. Nunc velit augue, scelerisque dignissim, lobortis et, aliquam in, risus. In eu eros. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Curabitur vulputate elit viverra augue. Mauris fringilla, tortor sit amet malesuada mollis, sapien mi dapibus odio, ac imperdiet ligula enim eget nisl. Quisque vitae pede a pede aliquet suscipit. Phasellus tellus pede, viverra vestibulum, gravida id, laoreet in, justo. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Integer commodo luctus lectus. Mauris justo. Duis varius eros. Sed quam. Cras lacus eros, rutrum eget, varius quis, convallis iaculis, velit. Mauris imperdiet, metus at tristique venenatis, purus neque pellentesque mauris, a ultrices elit lacus nec tortor. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent malesuada. Nam lacus lectus, auctor sit amet, malesuada vel, elementum eget, metus. Duis neque pede, facilisis eget, egestas elementum, nonummy id, neque.