**DOCUMENTATIERAPPORT**

onderdeel van de BACHELORPROEF

\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| **Analyse van REST-API technologieën** |
| Voor efficiënte en veilige toegang tot klantgegevens in externe applicaties |



|  |  |
| --- | --- |
| Bachelor | Toegepaste Informatica |
| Keuzetraject  Afstudeerrichting | Software Engineer |
| Academiejaar | 2023 - 2024 |
| Student | Lukas Olivier |
| Interne begeleider | Dieter Mourisse (Howest) |
| Externe promotor | Stefaan Missiean |

Melding van vertrouwelijkheid (enkel indien van toepassing)

*Indien dit documentatierapport vertrouwelijke informatie bevat, wordt de volgende tekst in verband met vertrouwelijkheid opgenomen. Indien niet, dan verwijder je deze pagina.*

Vertrouwelijk tot en met dd/mm/20yy

Belangrijk

Dit documentatierapport (dat onderdeel uitmaakt van de bachelorproef) bevat vertrouwelijke informatie en/of vertrouwelijke onderzoeksresultaten die toebehoren aan de Howest of aan derden. Dit rapport of enig onderdeel ervan mag op geen enkele wijze publiek gemaakt worden zonder de uitdrukkelijke schriftelijke voorafgaande toestemming vanwege Howest. Zo mag dit rapport onder geen voorwaarde door derden worden ingekeken of aan derden worden meegedeeld. Het is verboden om dit rapport te kopiëren of op eender welke manier te dupliceren. Indien de vertrouwelijke aard van dit rapport niet wordt gerespecteerd, kan dit onherstelbare schade veroorzaken aan Howest. Bovenstaande bepalingen zijn van kracht tot en met de embargodatum.

Toelating tot bruikleen

De auteur(s) geeft (geven) de toelating dit documentatierapport (onderdeel van de bachelorproef )voor consultatie beschikbaar te stellen en delen van dit rapport te kopiëren voor persoonlijk gebruik. Elk ander gebruik valt onder de bepalingen van het auteursrecht, in het bijzonder met betrekking tot de verplichting de bron uitdrukkelijk te vermelden bij het aanhalen van resultaten uit dit rapport.

26/03/2024

Woord vooraf

In het woord vooraf staan de gebruikelijke dankbetuigingen. Alle personen worden bedankt die meegeholpen hebben met het eindwerk.

De personen die de belangrijkste bijdrage hebben geleverd, worden als eerste bedankt. Schrijf de naam, functie en titel van personen correct.

Vermeld onderaan je naam, plaats en datum (facultatief). Een handtekening past hier niet.

Doordat het woord vooraf sterk persoonlijk is, wordt het vaak in de ik-vorm geschreven.

Inhoudsopgave

**Woord vooraf**

[1 Inleiding 5](#_Toc162015637)

[1.1 Algemeen 5](#_Toc162015638)

[1.2 Probleemstelling 5](#_Toc162015639)

[1.3 Onderzoeksvraag 6](#_Toc162015640)

[1.4 Wat is een API? 7](#_Toc162015641)

[1.4.1 API 7](#_Toc162015642)

[1.4.2 Web-API 7](#_Toc162015643)

[1.4.3 RESTful-API 7](#_Toc162015644)

[1.4.4 Belangrijke onderscheidingen: 7](#_Toc162015645)

[1.4.5 De voordelen van RESTful-API's: 8](#_Toc162015646)

[1.4.6 Conclusie: 8](#_Toc162015647)

[1.5 Experiment 8](#_Toc162015648)

[2 Experiment 9](#_Toc162015649)

[2.1.1 Demo Project: Student Management API 9](#_Toc162015650)

[2.1.2 Authenticatie: 9](#_Toc162015651)

[2.1.3 Operaties 10](#_Toc162015652)

[2.2 Express.js 11](#_Toc162015653)

[2.3 Django 13](#_Toc162015654)

[2.4 ASP.NET Core 15](#_Toc162015655)

[2.5 Ruby on Rails 16](#_Toc162015656)

[2.6 Samenvatting REST-API technologieën 17](#_Toc162015657)

[3 Conclusie 18](#_Toc162015658)

[AI Engineering Prompts 19](#_Toc162015659)

[4 Verwijzingen 20](#_Toc162015660)

[Code 20](#_Toc162015661)

[4.1.1 Express.js 20](#_Toc162015662)

[4.1.2 Django 20](#_Toc162015663)

[Bijlagen 21](#_Toc162015664)

[OpenAPI Specificatie 21](#_Toc162015665)

# Inleiding

## Algemeen

Deze bachelorproef, getiteld “Een grondige analyse van REST-API technologieën voor efficiënte en veilige toegang tot klantgegevens in externe applicaties,” is ontstaan tijdens mijn stage bij Integreat, een IT-serviceprovider in Waregem. Het onderzoek richt zich op het selecteren van een optimale REST-API voor klantgegevensintegratie, een essentieel aspect in de huidige digitale infrastructuur. Het primaire doelpubliek van deze proef zijn professionals en organisaties die streven naar een efficiënte en veilige manier om klantgegevens te beheren en te integreren in externe applicaties. De stagecontext heeft waardevolle praktijkervaring geboden, maar de nadruk van deze proef ligt op de algemene toepasbaarheid van de bevindingen voor het brede domein van software engineering.

## Probleemstelling

Dit onderzoek richt zich op het identificeren van de meest passende REST-API voor het bedrijf Integreat. Tot op heden kunnen klanten deze informatie enkel bekijken via klassieke import- en exportfunctionaliteiten.

De eigenaar van het probleem is het bedrijf Integreat, vertegenwoordigd door Stefaan Missiaen, de general manager van het bedrijf. Het is zijn behoefte aan een efficiënte en veilige manier voor klanten om gegevens op te halen die de aanleiding vormt voor dit onderzoek.

Er zijn verschillende REST-API-opties beschikbaar op de markt, maar geen enkele is tot nu toe geïmplementeerd.

Als dit probleem niet wordt opgelost, kunnen er diverse gevolgen optreden. In de eerste plaats zouden klanten van Integreat moeite kunnen ondervinden om op een efficiënte manier toegang te verkrijgen tot hun gegevens, wat kan leiden tot vertragingen en inefficiënties in hun eigen software-oplossingen. Daarnaast zou het ontbreken van een schaalbare REST-API-oplossing de toekomstige groei van Integreat en de tevredenheid van hun klanten kunnen beperken.

Het is daarom van essentieel belang om een geschikte REST-API te vinden die aan al deze eisen voldoet en waarmee een vlotte overgang van import-export functionaliteiten naar een REST-API mogelijk is.

## Onderzoeksvraag

De hoofdvraag die deze bachelor proef wil beantwoorden is: “Welke REST-API is het meest geschikt voor de toepassing van Integreat, om gegevens efficiënt en veilig op te vragen?”

Om antwoord te kunnen geven op de hoofdvraag, zijn de volgende deelvragen opgesteld:

1. Wat is een API?
2. Welke REST-API-opties zijn momenteel beschikbaar op de markt?
3. Hoe presteren deze REST-API-opties op het gebied van prestaties, productiviteit, veiligheid en ondersteuning?
4. Welke REST-API-optie biedt de beste balans tussen deze factoren en is het meest geschikt voor Integreat?

Deze deelvragen zullen helpen bij het grondig onderzoeken van het probleem en het vinden van de meest geschikte REST-API. Ze zullen ook zorgen voor een systematische aanpak van het onderzoek, wat zal bijdragen aan de kwaliteit en relevantie van de resultaten

## Wat is een API?

Voor we beginnen met het onderzoek moeten we natuurlijk verstaan wat een API is en doet.

### API

**API** staat voor Application Programming Interface. Het is een set van definities en protocollen die de communicatie tussen twee softwareapplicaties mogelijk maken. API's fungeren als tussenpersonen, waardoor applicaties **gegevens** kunnen **uitwisselen** en taken kunnen uitvoeren zonder de interne details van elkaar te kennen.

### ****Web-API****

**Web-API's** zijn een specifiek type API dat is ontworpen voor gebruik op het web. Ze gebruiken **HTTP als protocol** voor het verzenden en ontvangen van gegevens en zijn toegankelijk via URL's. REST-API's stellen ontwikkelaars in staat om webapplicaties te bouwen die met elkaar kunnen communiceren en gegevens kunnen delen met externe bronnen.

### RESTful-API

**RESTful-API's** (Representational State Transfer) zijn een subset van REST-API's die een set architectonische principes volgen om een uniforme en efficiënte manier van communiceren te garanderen. RESTful-API's gebruiken standaard HTTP-methoden (GET, POST, PUT, DELETE) en -codes om acties op resources te definiëren.

### ****Belangrijke onderscheidingen:****

* **Niet elke API is een REST-API:** Er bestaan API's die lokaal communiceren tussen verschillende onderdelen van dezelfde applicatie of die gebruikmaken van andere protocollen dan HTTP.
* **Niet elke REST-API is een RESTful-API:** Sommige REST-API's gebruiken andere architectonische stijlen of bieden minder strikte richtlijnen voor communicatie.

### ****De voordelen van RESTful-API's:****

* **Eenvoudig te gebruiken:** De standaard HTTP-methoden en -codes zijn universeel bekend en eenvoudig te begrijpen.
* **Lichtgewicht:** RESTful-API's verbruiken minder bandbreedte en resources dan andere API-stijlen.
* **Schaalbaar:** RESTful-API's zijn goed uit te breiden en kunnen eenvoudig worden aangepast aan nieuwe eisen.
* **Toekomstbestendig:** De REST-architectuur is al decennia lang in gebruik en heeft bewezen robuust en flexibel te zijn.

### ****Conclusie:****

API's zijn essentiële tools voor het bouwen van moderne softwareapplicaties. REST-API's en RESTful-API's vormen een krachtige combinatie die efficiënte communicatie tussen applicaties op het web mogelijk maakt. Inzicht in de werking en voordelen van deze technologieën is van cruciaal belang voor ontwikkelaars die webapplicaties willen bouwen die naadloos met elkaar kunnen communiceren.

In dit document zullen we ons focussen op **RESTful Web-API's**, die we voor de beknoptheid verder zullen afkorten naar **REST-API's**. We bespreken de werking van REST-API's, de voordelen die ze bieden en de verschillende frameworks die beschikbaar zijn voor hun ontwikkeling.

## Experiment

Dit onderzoek combineert kwalitatief en kwantitatief onderzoek. Het kwalitatieve deel richt zich op de behoeften van scholen betreffende gegevensverzameling via een REST-API. Het kwantitatieve deel evalueert en vergelijkt REST-API opties op criteria zoals veiligheid en efficiëntie. Data worden verzameld via literatuur- en online onderzoek naar beschikbare REST-API’s, bestaande uit technische specificaties en prestatiegegevens. Analyse en vergelijking van de data zullen de meest geschikte REST-API identificeren, waarbij elke optie wordt geëvalueerd en vergeleken op basis van criteria.

# Experiment

Om een grondige vergelijking tussen verschillende API's mogelijk te maken, is het essentieel om uitgebreid onderzoek te doen naar de diverse mogelijkheden. Met als doel een breed scala aan perspectieven te verkrijgen, streef ik ernaar een vergelijkbare demo-API te repliceren in verschillende programmeertalen. Voor elke programmeertaal zal ik het meest populaire en/of effectieve framework gebruiken om een API te ontwikkelen. Daarna documenteerde ik telkens mijn bevindingen.

Gezien scholen een aanzienlijke klantengroep zijn voor Integreat, zal ik als voorbeeld een Student Management API ontwikkelen.

### Demo Project: Student Management API

De functionaliteiten van de API zijn vastgelegd in de Swagger OpenAPI-specificatie. Uitgebreide details over de beschikbare handelingen, inclusief authenticatie met JSON Web Tokens (JWT), zijn te vinden in de bijlagen. Hieronder volgt een samenvatting van enkele belangrijke handelingen.

### Authenticatie:

De REST-API maakt gebruik van JWT-tokens voor authenticatie. Op basis van deze tokens wordt alleen informatie teruggegeven waar de gebruiker toegang toe heeft. Bijvoorbeeld, het opvragen van een lijst met studenten zal alleen studenten opleveren die aan de school van de gebruiker zijn gekoppeld.

### Operaties

Hieronder volgt een lijst van endpoints die elke REST-API zal bevatten:

* **Lijst van scholen:** Hiermee kan de gebruiker de lijst van scholen opvragen die hij of zij bezit.
* **Lijst van studenten per school**: Hiermee kan de gebruiker een lijst van studenten opvragen die zijn ingeschreven bij een specifieke school**.**
* **Lijst van niet-ingeschreven studenten**: Dit endpoint maakt het mogelijk om een lijst op te vragen van studenten die niet zijn ingeschreven bij een school.
* **Inschrijven bij een school:** Hiermee kunnen studenten worden ingeschreven bij een specifieke school waarvan de gebruiker eigenaar is.
* **Uitschrijven bij een school:** Dit endpoint maakt het mogelijk voor de gebruiker om studenten uit te schrijven die zijn ingeschreven bij een school die aan hem/haar is gekoppeld.
* **Toevoegen van nieuwe studenten:** Gebruikers kunnen nieuwe studenten toevoegen met alle benodigde informatie.
* **Verwijderen van studenten:** Dit endpoint biedt de mogelijkheid om studenten te verwijderen uit het systeem.

Voor een gedetailleerde beschrijving van de API-handelingen, verwijzen we naar de Swagger OpenAPI-specificatie in de bijlagen.

## Express.js

#### Waarom

Express.js is een robuust en flexibel framework voor het ontwikkelen van webapplicaties en API's binnen Node.js. Met een minimalistische aanpak biedt het een intuïtieve manier om routes te definiëren, middleware toe te passen en HTTP-verzoeken te verwerken. Hier zijn enkele redenen waarom Express.js een uitstekende keuze is voor het bouwen van REST-API's:

#### Ondersteuning en Documentatie:

Express.js wordt ondersteund door uitgebreide en beknopte documentatie, inclusief tutorials, API-referenties en voorbeeldcode. Deze bronnen zijn waardevol voor ontwikkelaars om snel aan de slag te gaan en complexe problemen op te lossen. Bovendien is er een actieve community beschikbaar via forums en platforms zoals Stack Overflow, waardoor ontwikkelaars gemakkelijk hulp en ondersteuning kunnen vinden.

#### Veiligheid:

Express.js biedt verschillende libraries om de beveiliging te waarborgen, zoals het hashen van wachtwoorden met Bcrypt, autorisatie met JSON Web Tokens (JWT) en validatie met JOI. Gevoelige informatie wordt apart gehouden in een .env-bestand, terwijl extra bescherming tegen aanvallen wordt geboden door middel van de Helmet library.

#### Prestatie:

Express.js staat bekend om zijn hoge prestaties, zelfs onder zware belasting. Het framework is geoptimaliseerd om efficiënt te werken en kan snel reageren op HTTP-verzoeken, waardoor het ideaal is voor het bouwen van schaalbare API's die grote hoeveelheden verkeer aankunnen.

#### Ontwikkelaarsproductiviteit:

Dankzij de eenvoudige syntax en de volwassenheid van het framework kunnen ontwikkelaars snel en efficiënt werken. Ik kon het volledige REST-API demo project een binnen één dag opzetten, inclusief de benodigde beveiligingsmechanismen, CRUD operaties en user management. Bovendien is er een overvloed aan informatie en voorbeeldcode beschikbaar op het internet, waardoor het gemakkelijk is om oplossingen te vinden voor specifieke problemen.

#### Conclusie:

Express.js een uitstekende keuze voor het ontwikkelen van REST-API's. Het framework biedt een balans tussen snelheid, flexibiliteit, veiligheid en ondersteuning, waardoor het geschikt is voor een breed scala aan projecten. Met zijn bewezen prestaties en uitgebreide gemeenschap is Express.js een betrouwbare keuze voor zowel kleine als grote projecten.

## Django

#### Waarom

Django, gebouwd op Python, is een krachtig framework dat een breed scala aan mogelijkheden biedt voor het ontwikkelen van webapplicaties en REST-API's. Hoewel het framework meer dan alleen een API-ondersteuning biedt en ook geschikt is voor front-end ontwikkeling, is het nog steeds een waardevolle optie, zelfs voor demo-doeleinden, dankzij zijn uitgebreide functionaliteit en solide documentatie.

#### Ondersteuning / documentatie

De documentatie van Django is modern en van hoge kwaliteit, waardoor ontwikkelaars snel aan de slag kunnen en complexe problemen kunnen oplossen. Bovendien is er een bloeiende community beschikbaar voor ondersteuning via forums. Dit maakt het gemakkelijk om hulp te vinden bij eventuele obstakels tijdens de ontwikkeling.

#### Ontwikkelaarsproductiviteit

Het opzetten en uitbreiden van een API met Django verloopt over het algemeen vlot en moeiteloos. De framework biedt een krachtige set tools en functionaliteiten die ontwikkelaars in staat stellen om snel aan de slag te gaan. Echter, ik ben minder enthousiast over de mappenstructuur van Django, aangezien er redelijk veel functionaliteit geconcentreerd is in dezelfde bestanden, wat tot enige onduidelijkheid kan leiden tijdens het ontwikkelproces.

Daarnaast vereist het integreren van een bestaande database en tabellen wat extra configuratie en aanpassingen, wat mogelijk de ontwikkeltijd kan verlengen. Echter, wanneer je te maken hebt met een project waarvoor een nieuwe database en tabellen nodig zijn, verloopt dit proces zeer vlot met behulp van migrations.

#### Prestatie

De prestaties van Django zijn over het algemeen solide. Django snel en efficiënt reageren op HTTP-verzoeken, waardoor het een geschikte keuze is voor het bouwen van schaalbare API's die grote hoeveelheden verkeer kunnen verwerken.

#### Veiligheid

Django biedt ingebouwde beveiligingsmechanismen voor authenticatie, autorisatie en gegevensvalidatie. Dit omvat bescherming tegen veelvoorkomende aanvallen zoals XSS, CSRF en SQL-injectie. Echter, vanwege het gebrek aan veel third-party libraries voor bijvoorbeeld validatie, kan het zijn dat ontwikkelaars meer handmatig werk moeten verrichten om aanvullende beveiligingsmaatregelen te implementeren.

#### Conclusie:

Django is een krachtig framework voor het ontwikkelen van webapplicaties en REST-API's, met solide documentatie en een brede community voor ondersteuning. Hoewel het opzetten van een API over het algemeen vlot verloopt, kan de mappenstructuur enige onduidelijkheid veroorzaken. Integratie met bestaande databases kan extra configuratie vereisen, hoewel het migratieproces soepel verloopt voor nieuwe databases. De prestaties van Django zijn solide en het biedt ingebouwde beveiligingsmechanismen, maar het gebrek aan third-party libraries kan ontwikkelaars dwingen om meer handmatig werk te verrichten voor aanvullende beveiligingsmaatregelen. Over het algemeen blijft Django een waardevolle keuze voor het ontwikkelen van zowel kleine als grote projecten.

## ASP.NET Core

#### Waarom

Waarom dit framework?

#### Ondersteuning / documentatie

Kijk naar de documentatiekwaliteit en community-ondersteuning voor het oplossen van problemen.

#### Ontwikkelaarsproductiviteit

Beoordeel hoe snel en gemakkelijk het is om de API op te zetten / uit te breiden

#### Veiligheid

Onderzoek de ingebouwde beveiligingsmechanismen zoals authenticatie, autorisatie en gegevensvalidatie. Beoordeel de mogelijkheden voor het beschermen tegen veelvoorkomende aanvallen zoals XSS, CSRF en SQL-injectie.

#### Prestatie

Meet de prestaties van de API's onder verschillende belastingniveaus.

## Ruby on Rails

#### Waarom

Waarom dit framework?

#### Ondersteuning / documentatie

Kijk naar de documentatiekwaliteit en community-ondersteuning voor het oplossen van problemen.

#### Ontwikkelaarsproductiviteit

Beoordeel hoe snel en gemakkelijk het is om de API op te zetten / uit te breiden

#### Veiligheid

Onderzoek de ingebouwde beveiligingsmechanismen zoals authenticatie, autorisatie en gegevensvalidatie. Beoordeel de mogelijkheden voor het beschermen tegen veelvoorkomende aanvallen zoals XSS, CSRF en SQL-injectie.

#### Prestatie

Meet de prestaties van de API's onder verschillende belastingniveaus.

## Samenvatting REST-API technologieën

*GROTE TABEL MET CHECKS*

*Hashing? Validatie? Sql injection? Goeie documentatie?*

*Persoonlijke rankschikking ease of use*

*Rangschikking prestatie mbv Ddosify*

# Conclusie

In het conclusie-gedeelte formuleer je het antwoord op de onderzoeksvraag.

Hierin komen geen nieuwe resultaten aan bod die je nog niet eerder hebt aangehaald. Gebruik hier geen subsecties.

Sluit op het einde af met een krachtige slotzin die jouw conclusie in een zin kort samenvat.

# AI Engineering Prompts

Tijdens het schrijven van dit document heb ik gebruik gemaakt van de volgende algemene prompt om teksten te verbeteren en aantrekkelijker te maken voor de lezers:

*“Kun je deze tekst verbeteren en aantrekkelijker maken voor de lezers in ongeveer X woorden?”*

Deze prompt is bedoeld om de AI te vragen om een gegeven tekst te herformuleren, te verbeteren en aantrekkelijker te maken binnen een bepaald woordlimiet. Het is belangrijk op te merken dat de exacte woordlimiet (X) kan variëren afhankelijk van de specifieke tekst en de context.

# Verwijzingen

*Integreat*. (2024, 02 16). Opgehaald van Integreat: https://www.integreat.be/

Missiaen, S. (2024, 02 16). Interview over Integreat. (L. Olivier, Interviewer)

Wat is een api, web-api & REST-api

* <https://nl.wikipedia.org/wiki/Application_programming_interface>
* <https://en.wikipedia.org/wiki/REST>
* <https://www.redhat.com/en/topics/api/what-is-a-rest-api>
* <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Introduction>
* <https://www.w3schools.com/js/js_api_intro.asp>

## Code

### Express.js

Setup / best practices

<https://mahmoud-kassem.medium.com/how-to-build-an-api-with-node-js-express-and-typescript-2024-extended-part-1-6-f65df183dbc5>

Security

<https://www.youtube.com/watch?v=Tw5LupcpKS4>

<https://www.youtube.com/watch?v=RtLAwnYJOyQ>

### Django

<https://www.youtube.com/watch?v=t-uAgI-AUxc>

[https://youtu.be/llrIu4Qsl7c?si=CenP\_g3JVmEq2jq7](https://youtu.be/llrIu4Qsl7c?si=CenP_g3JVmEq2jq7%0c)

# Bijlagen

### OpenAPI Specificatie

<https://github.com/LukasOlivier/bap/blob/main/documentation/openapi.yaml>