

Ervaringsweek3-Blog Central

We gaan de volgende tools en frameworks gebruiken tijdens dit project:

- **Scrum**
- **Trello (taakverdeling)**
- **Github (werkt met verschillende branches)**
- **Java**
- **Intellij**
- **Mysql**
- **Spring Boot**
- **Postman**

We hebben 3 groepen:

Groep 1 -> Eva, Tamara, Chaima, Viktoriia

Groep 2 -> Annelies, Richard, Maxime, Oleksandr

Groep 3 -> Ira, Bram, Haydar, Dmytro

Allereerst gaan jullie je team opsplitsen in rollen. Je laat me via een bericht weten wie welke rol zal opnemen, dus wie is product-owner, scrummaster en wie zijn de developers.

1-Scrum toepassen voor dit project:

- In je team gaat iemand de user stories maken. Voor dit project zal het de product-owner zijn. Je overlegt met je teamgenoten wat de functies gaan zijn en vult die in op het Trello bord.
- Eén lid van het team zal de scrum master zijn. - De rest zijn de developers. Iedereen codeert mee, ook al ben je scrum master of product-owner. Je gaat met je team een Trello copy maken. De minimale implementatie waar je voor moet zorgen is:

- De taken onderverdelen met bijvoorbeeld: to-do, in progress, done. Elke taak komt in de juiste kolom (minimale 3 verschillende statussen).

- Zorgen dat je de taken kunt veranderen van status, bijvoorbeeld: van to-do naar in-progress en ook weer terug naar to-do.

2-Vergeet niet om een analyse van het project te maken, inclusief klassendiagrammen, flowcharts en sequence diagrammen.

3-Laatste deel van de opdracht is een presentatie maken in teamvorm. Hier zullen jullie jullie project demonstreren en key features tonen:

- Live demo

- Waarom zou ik dit product kopen?

- Lengte van de presentatie minimaal 20 min en maximaal 30 min.

- Iedereen moet een deel presenteren

1. Datamodel

De backend bevat minimaal drie hoofd-entiteiten. Dit is de basis voor de database en wordt gebruikt bij het opzetten van de API.

1.1 User

- id (PK)
- firstName
- lastName
- username (uniek, verplicht)
- email
- password (gehasht)
- avatarUrl (optioneel)
- adresgegevens: street, houseNr, city, zip (optioneel)
- lijst van blogposts
- lijst van commentaren

1.2 BlogPost

- id (PK)
- title

- content
- createdAt (datum/tijd)
- likes (aantal)
- author (User-relatie)
- list of comments

1.3 Comment

- id (PK)
- content
- createdAt
- author (User-relatie)
- blogPost (BlogPost-relatie)

2. Projectomschrijving

In dit project bouw je een volledige backend voor een blogplatform: **Blog Central**. Gebruikers kunnen blogposts lezen, schrijven, liken en becommentariëren.

De volledige implementatie gebeurt in **Spring Boot** met **MySQL** als database.

Er is **geen frontend vereist**: alle functionaliteit moet aantoonbaar werken via **Postman**. Jullie mogen optioneel een frontend toevoegen (bijvoorbeeld Angular), maar dit heeft geen invloed op de beoordeling.

Het doel van dit project is aantonen dat je een **professionele, veilige, schaalbare en REST-compatibele backend** kunt ontwikkelen.

3. Gebruikersrollen

3.1 Bezoekers (niet-ingelogd)

Bezoekers kunnen:

- Blogposts lezen

- Blogposts zoeken
- Registreren
- Inloggen

3.2 Auteurs (ingelogd)

Auteurs kunnen alles wat bezoekers kunnen, plus:

- Uitloggen
- Hun eigen profiel beheren (bekijken / aanpassen / verwijderen)
- Blogposts aanmaken
- Eigen blogposts bewerken of verwijderen
- Commentaren plaatsen
- Eigen commentaren bewerken of verwijderen
- Commentaren op eigen blogposts modereren (verwijderen)

4. Backend Functionaliteiten

Je implementeert alle functionaliteit volgens moderne REST API principes:

- Validatie
- Veiligheid
- JWT-authenticatie
- Autorisatie op basis van eigenaarschap
- Juiste foutafhandeling

4.1 User Management

Registratie

- Verplichte velden: firstName, lastName, username, email, password, retypePassword
- Username moet uniek zijn
- Password en retypePassword moeten gelijk zijn

Login

- Inloggen met username + password
- Respons bevat JWT-token

Authenticatie & Autorisatie

- JWT-token nodig voor beveiligde endpoints
- Ingelogde gebruikers mogen niet opnieuw registreren of inloggen

Profielbeheer

- Gebruikers kunnen hun eigen profiel opvragen
- Aanpassen van alle velden behalve username
- Profiel verwijderen resulteert in automatische logout

Keep me logged in

- Token met langere vervaltijd mogelijk

4.2 Blogposts

Blogpost aanmaken

- Alleen ingelogde gebruikers

Blogpost wijzigen & verwijderen

- Alleen de auteur van de post

Blogposts ophalen

Toegankelijk voor iedereen, met:

- Sortering: newest (default), oldest, popular
- Paginatie: standaard 6 per pagina

Blogpost details

Bevat:

- Titel, inhoud
- Auteurinfo
- Likes
- Alle commentaren

Like-functie

- Alleen ingelogde gebruikers

4.3 Commentaren

Comment toevoegen

- Alleen ingelogde gebruikers
- Niet ingelogd → 401 Unauthorized

Comment bewerken / verwijderen

- Alleen de auteur zelf

Moderatie

- De auteur van een blogpost mag alle commentaren op die post verwijderen

Commentaren ophalen

- Automatisch zichtbaar bij blogpost details

5. API Endpoints

Alle beveiligde endpoints vereisen:

Authorization: Bearer <token>

5.1 USER ENDPOINTS

- POST **/users/register** – registreren
- POST **/users/login** – inloggen
- GET **/users/{id}** – profiel ophalen
- PUT **/users/{id}** – profiel aanpassen
- DELETE **/users/{id}** – profiel verwijderen

5.2 BLOGPOST ENDPOINTS

- GET **/blogposts?sort=&page=&size=** – lijst blogposts
- GET **/blogposts/{postId}** – details blogpost
- POST **/blogposts** – nieuwe blogpost
- PUT **/blogposts/{postId}** – blogpost bewerken
- DELETE **/blogposts/{postId}** – blogpost verwijderen
- POST **/blogposts/{postId}/like** – blogpost liken

5.3 COMMENT ENDPOINTS

- POST **/blogposts/{postId}/comments** – comment toevoegen
- PUT **/comments/{commentId}** – comment bewerken
- DELETE **/comments/{commentId}** – comment verwijderen

5.4 Zoekfunctionaliteit

- GET **/search?q=zoekterm** – zoeken op titel, gebruiker of tag

5.5 Statistieken

- GET **/stats/online** – aantal online gebruikers
- GET **/stats/visitors** – totaal aantal bezoekers

6. Autorisatieoverzicht

Functionaliteit	Wie mag dit?
Registreren	Niet-ingelogd
Inloggen	Niet-ingelogd
Blogposts bekijken	Iedereen
Blogpost detail	Iedereen
Blogpost aanmaken	Ingelogd
Blogpost bewerken/verwijderen	Auteur van de blogpost
Comment toevoegen	Ingelogd
Comment bewerken/verwijderen	Auteur van het commentaar
Comment verwijderen (moderatie)	Auteur van de blogpost
Profiel aanpassen/verwijderen	Eigen gebruiker

7. Testen met Postman

Cursist toont werking aan via Postman :

- Registreren
- Inloggen (token ontvangen)
- Uitvoeren van beveiligde requests met geldige/ongeldige token
- Blogposts CRUD
- Commentaren CRUD
- Sorteren & pagineren
- Edge cases testen (403 Forbidden → bij gebrek aan eigenaarschap)
- Zoeken

8. Doel van het ervaringsweekproject

Na afronding toont de cursist aan dat hij/zij:

- Volledige REST API kan ontwerpen
- CRUD-operaties correct implementeert
- JWT-authenticatie beheerst
- Autorisatie correct afdwingt
- Relationale database kan modelleren
- API's correct en professioneel kan testen
- Backend-framework (Spring Boot) volgens best practices toepast

Veel succes.

CHECKLIST

1. Projectstructuur & Setup

Item	JA	NEE
Spring Boot project start correct op zonder fouten	<input type="checkbox"/>	<input type="checkbox"/>
Juiste projectstructuur: controller, service, repository, model	<input type="checkbox"/>	<input type="checkbox"/>
MySQL-database is correct geconfigureerd (application.properties)	<input type="checkbox"/>	<input type="checkbox"/>
Database wordt automatisch aangemaakt (JPA/Hibernate)	<input type="checkbox"/>	<input type="checkbox"/>
Password hashing correct toegepast (BCrypt of Argon2)	<input type="checkbox"/>	<input type="checkbox"/>

2. Datamodel

Item	JA	NEE
User-entiteit bevat verplichte velden + unieke username	<input type="checkbox"/>	<input type="checkbox"/>
BlogPost-entiteit bevat titel, content, author, timestamp, likes	<input type="checkbox"/>	<input type="checkbox"/>
Comment-entiteit gekoppeld aan User + BlogPost	<input type="checkbox"/>	<input type="checkbox"/>
Relaties correct geconfigureerd (OneToMany, ManyToOne, Cascade)	<input type="checkbox"/>	<input type="checkbox"/>

Lazy/Fetch types correct toegepast

☐ ☐

3. User Management

Item	JA	NEE
Registratie werkt en valideert alle velden	<input type="checkbox"/>	<input type="checkbox"/>
Dubbele gebruikersnamen worden geweigerd	<input type="checkbox"/>	<input type="checkbox"/>
Wachtwoord en herhaal-wachtwoord worden vergeleken	<input type="checkbox"/>	<input type="checkbox"/>
Login werkt en geeft JWT-token terug	<input type="checkbox"/>	<input type="checkbox"/>
JWT wordt correct gebruikt bij beveiligde endpoints	<input type="checkbox"/>	<input type="checkbox"/>
Ingelogde gebruikers kunnen hun eigen profiel opvragen	<input type="checkbox"/>	<input type="checkbox"/>
Gebruikers kunnen hun profiel aanpassen (zonder username)	<input type="checkbox"/>	<input type="checkbox"/>
Gebruikers kunnen hun profiel verwijderen	<input type="checkbox"/>	<input type="checkbox"/>
Keep-me-logged-in werkt (optie: langere token geldigheid)	<input type="checkbox"/>	<input type="checkbox"/>

4. BlogPost Functionaliteiten

Item	JA	NEE
Blogpost aanmaken werkt (alleen ingelogd)	<input type="checkbox"/>	<input type="checkbox"/>
Blogpost bewerken werkt (alleen auteur)	<input type="checkbox"/>	<input type="checkbox"/>
Blogpost verwijderen werkt (alleen auteur)	<input type="checkbox"/>	<input type="checkbox"/>
Blogposts ophalen werkt voor iedereen	<input type="checkbox"/>	<input type="checkbox"/>
Sortering werkt (newest / oldest / popular)	<input type="checkbox"/>	<input type="checkbox"/>
Paginatie werkt correct (default 6 per pagina)	<input type="checkbox"/>	<input type="checkbox"/>
Detail-endpoint toont likes, auteur en commentaren	<input type="checkbox"/>	<input type="checkbox"/>
Like-functie werkt (alleen ingelogd)	<input type="checkbox"/>	<input type="checkbox"/>

5. Comment Functionaliteiten

Item	JA	NEE
Comment toevoegen werkt (alleen ingelogd)	<input type="checkbox"/>	<input type="checkbox"/>
Comment bewerken werkt (alleen eigen comment)	<input type="checkbox"/>	<input type="checkbox"/>
Comment verwijderen werkt (alleen eigen comment)	<input type="checkbox"/>	<input type="checkbox"/>

Moderatie: auteur blogpost kan comments verwijderen	<input type="checkbox"/>	<input type="checkbox"/>
Comments worden opgehaald samen met blogpost details	<input type="checkbox"/>	<input type="checkbox"/>

6. Zoekfunctionaliteit

Item	JA	NEE
Zoeken werkt via /search?q=	<input type="checkbox"/>	<input type="checkbox"/>
Zoekresultaten tonen blogposts op basis van titel / auteur / tag	<input type="checkbox"/>	<input type="checkbox"/>
Sortering binnen zoekresultaten werkt (same as /blogposts)	<input type="checkbox"/>	<input type="checkbox"/>

7. Statistieken

Item	JA	NEE
/stats/online geeft correct aantal ingelogde gebruikers	<input type="checkbox"/>	<input type="checkbox"/>
/stats/visitors toont totaal aantal bezoekers	<input type="checkbox"/>	<input type="checkbox"/>

8. Postman Testing

Item	JA	NEE
Alle endpoints getest via Postman	<input type="checkbox"/>	<input type="checkbox"/>
Valid token → toegang tot beveiligde endpoints	<input type="checkbox"/>	<input type="checkbox"/>
Ongeldige token → 401 Unauthorized	<input type="checkbox"/>	<input type="checkbox"/>
Niet-eigen post/comment → 403 Forbidden	<input type="checkbox"/>	<input type="checkbox"/>
Register/Login werkt met correcte responses	<input type="checkbox"/>	<input type="checkbox"/>
CRUD-acties getest voor Users, Blogposts, Comments	<input type="checkbox"/>	<input type="checkbox"/>
Edge cases getest (lege velden, invalid input, forbidden actions)	<input type="checkbox"/>	<input type="checkbox"/>
Export van Postman collectie aanwezig als bewijs	<input type="checkbox"/>	<input type="checkbox"/>

9. Eindcontrole

Item	JA	NEE
Alle functionaliteiten zijn geïmplementeerd	<input type="checkbox"/>	<input type="checkbox"/>
Code is duidelijk, schoon en professioneel	<input type="checkbox"/>	<input type="checkbox"/>
Project start zonder foutmeldingen	<input type="checkbox"/>	<input type="checkbox"/>
Database werkt correct en consistent	<input type="checkbox"/>	<input type="checkbox"/>
Documentatie (READme/handleiding) is aanwezig	<input type="checkbox"/>	<input type="checkbox"/>