04. Opdracht Servlets

## Stappen

Hier is een "doe-het-zelf" opdracht voor de studenten:

1. Pas de BooksServlet aan om het bijwerken van boeken via PUT-verzoeken te ondersteunen.
2. Implementeer validatie om ervoor te zorgen dat alle vereiste velden aanwezig zijn in POST- en PUT-verzoeken en dat de boek-ID's geldig zijn.
3. Voeg foutmeldingen toe aan de API-responses als er velden ontbreken of ongeldig zijn.
4. Implementeer een nieuwe servlet genaamd AuthorsServlet die een REST API implementeert voor het beheren van auteurs.
5. Maak een nieuw Author-object aan met velden zoals naam, leeftijd en nationaliteit.
6. Implementeer endpoints voor GET-, POST-, PUT- en DELETE-verzoeken om auteurs toe te voegen, bij te werken, op te halen en te verwijderen.
7. Implementeer validatie om ervoor te zorgen dat alle vereiste velden aanwezig zijn in POST- en PUT-verzoeken en dat de auteurs-ID's geldig zijn.
8. Voeg foutmeldingen toe aan de API-responses als er velden ontbreken of ongeldig zijn.
9. Integreer de BooksServlet en de AuthorsServlet om de gegevens van de boeken en de auteurs in één API te beheren.
10. Test de API grondig met behulp van een REST-client zoals Postman of cURL om er zeker van te zijn dat het correct werkt en fouten worden afgehandeld.

Deze "doe-het-zelf" opdracht geeft de je de kans om meer te leren over het implementeren van RESTful API's met behulp van Java Servlets en om hun vaardigheden te oefenen door middel van een praktische opdracht.

## Stapsgewijze Handleiding

Hieronder vindt u een stapsgewijze handleiding voor het maken van een nieuw project met JPA, Hibernate en Java Servlet:

1. Maak een nieuw project aan in IntelliJ IDEA.
2. Maak een nieuwe map aan genaamd src.
3. Maak een submap aan genaamd main.
4. Maak een submap aan genaamd java.
5. Maak een nieuwe Java-klasse genaamd JpaUtil. Deze klasse bevat de logica voor het opzetten van de JPA EntityManagerFactory.
6. Maak een nieuwe Java-klasse genaamd EntityManagerHelper. Deze klasse bevat de logica voor het verkrijgen van een EntityManager voor het uitvoeren van CRUD-acties.
7. Maak een nieuwe Java-klasse genaamd User. Deze klasse bevat de definitie van een gebruiker en zijn eigenschappen zoals ID, naam, e-mailadres en wachtwoord.
8. Maak een nieuwe Java-klasse genaamd UserDAO. Deze klasse bevat de logica voor het uitvoeren van CRUD-acties op de gebruikersentiteit met behulp van de EntityManager.
9. Maak een nieuwe Java-klasse genaamd HibernateUtil. Deze klasse bevat de logica voor het opzetten van de Hibernate SessionFactory.
10. Maak een nieuwe Java-klasse genaamd UserServlet. Deze klasse is verantwoordelijk voor het ontvangen van de HTTP-verzoeken en het doorgeven van de gegevens aan de UserDAO voor verwerking.
11. Implementeer de doGet-methode om alle gebruikers op te halen en terug te sturen als JSON-gegevens.
12. Implementeer de doPost-methode om een nieuwe gebruiker toe te voegen met behulp van de JSON-gegevens die zijn ontvangen van de client.
13. Implementeer de doPut-methode om een bestaande gebruiker bij te werken met behulp van de JSON-gegevens die zijn ontvangen van de client.
14. Implementeer de doDelete-methode om een gebruiker te verwijderen met behulp van de JSON-gegevens die zijn ontvangen van de client.
15. Maak een nieuwe Java-klasse genaamd UserDTO. Deze klasse bevat de definitie van de gegevensoverdrachtsobjecten die worden gebruikt om de gegevens tussen de servlet en de DAO-klasse te verzenden.
16. Implementeer de conversie van User-entiteit naar UserDTO-object en vice versa.
17. Voeg de vereiste JPA- en Hibernate-afhankelijkheden toe aan het pom.xml-bestand.
18. Configureer de JPA- en Hibernate-instellingen in het persistence.xml-bestand.
19. Voeg een nieuwe servlet-mapping toe aan het web.xml-bestand om de UserServlet te koppelen aan de /users-URI.
20. Bouw en deploy de applicatie naar een webcontainer zoals Tomcat en test deze uitvoerig met behulp van een REST-client zoals Postman of cURL om er zeker van te zijn dat het correct werkt en fouten worden afgehandeld.

## Verwachtingen

Deze opdracht daagt studenten uit om een volledig functionerende webtoepassing te bouwen met behulp van JPA, Hibernate en Java Servlets. Het vereist ook dat ze bekend zijn met het implementeren van een RESTful API en het gebruik van JSON-gegevens om de gegevens tussen de client en de server over te dragen.

Naast het implementeren van de basisfunctionaliteit van het toevoegen, bijwerken, ophalen en verwijderen van gebruikers, kunnen de studenten ook extra functionaliteit toevoegen, zoals authenticatie en autorisatie, het gebruik van aangepaste query's om gegevens op te halen en andere CRUD-operaties uit te voeren op andere entiteiten.

Studenten kunnen ook worden uitgedaagd om de applicatie verder te schalen en te optimaliseren door het gebruik van caching, indexing en load balancing. Dit kan helpen om de prestaties te verbeteren en de reactietijden van de applicatie te verkorten, vooral bij het verwerken van grote hoeveelheden gegevens of wanneer er meerdere gelijktijdige verzoeken worden verwerkt.

## Samenvatting

Al met al kan deze opdracht een uitdagende en interessante manier zijn om de vaardigheden van studenten op het gebied van webontwikkeling te verbeteren en hen te laten werken aan een project dat nuttig kan zijn voor het opbouwen van een professioneel portfolio.