

Programmeeropdracht

Creëer, op basis van een gelaagde architectuur, een Restful-backend met behulp van JEE. De applicatie moet een Restfull API aanbieden en bruikbaar zijn met een testclient.

Daarbij moet een opleverdocument worden gemaakt, met daarin o.a. een packagediagram en een deploymentdiagram.

De benodigde onderdelen:

- De volledige opdrachtomschrijving: [OpdrachtSpotitube.pdf](#)
- De te implementeren Restfull API: <https://github.com/HANICA-DEA/spotitube>
- De testclient: <http://ci.icaprojecten.nl/spotitube/>

User stories (learning objectives)

Software Design

As a student, I need to learn software patterns and design notations so I can use a shared language to discuss design decisions with my team mates.

Acceptatiecriteria:

- Kent het layer pattern en kan een eenvoudig REST-backend ontwikkelen op basis van een casus en API-specificatie.
- Kent de verschillen en overeenkomsten tussen layers (logisch model) en packages (fysiek model).
- Kent de verschillende categorieën waarin EAA patterns worden ingedeeld.
- Kan een opleverdocument maken dat
 - voldoet aan de ICA stijlkaart voor documenten
 - een package diagram bevat, voorzien van uitleg over de gemaakte keuzes.
 - een deployment diagram bevat, voorzien van uitleg over de gemaakte keuzes.

JEE

As a student, I need to learn the JEE platform so I can build scalable web applications.

Acceptatiecriteria:

- Kent de volgende JEE APIs en kan deze toepassen in een eenvoudige back-end applicatie die runt in TomEE: CDI, JDBC en JAX-RS.
- Kent de volgende JEE APIs en kan deze plaatsen in het layer model: JAX-RS, JDBC, CDI.

JEE - Data Source Layer

As a student, I need to learn JDBC so I can learn how to implement a Data Source Layer and access databases.

Acceptatiecriteria:

- Kan met JDBC connecties opbouwen naar een relationele database, gebruik makend van een configuratiebestand met connectiegegevens.
- Kan met JDBC het Data Mapper pattern implementeren voor eenvoudige CRUD-operaties voor een handvol tabellen.

JEE - Domain Layer

As a student, I need to learn how to build REST-ful applications with Jersey so I can learn how to implement a Service Layer on top of a Domain Model.

Acceptatiecriteria:

- Kent de vier verschillende patterns om domein logica te organiseren: Transaction Script, Domain Model, Table Module, Service Layer
- Kan op basis van tekstuele beschrijving van het domein het Data Transfer Object pattern implementeren in Java.

Dependency Injection

As a student, I need to learn Dependency Injection so I can lower the coupling between layers and improve testability.

Acceptatiecriteria:

- Kent de relatie tussen het Dependency Inversion Principle (DIP uit SOLID) en de techniek van Dependency Injection. Kent minimaal 1 alternatief voor Dependency Injection waarmee je toch kunt voldoen aan het DIP.
- Kan gebruik makend van CDI de koppeling verlagen tussen de verschillende layers in een JEE applicatie.
- Kan middels een unit-test aantonen dat de ontkoppeling van lagen geslaagd is.

Beschikbare leermiddelen

Architectural Layers & Patterns

Bron	Hoofdstuk	Toetsstof
Sheet	02. Layer pattern.pptx 06. Domain Layer.pptx 09. Deployment diagrams.pptx 10. Package diagrams.pptx	Ja
Applying UML and Patterns	H34: Logical Architecture Refinement	Ja
Applying UML and Patterns	H36: Package Design	Ja
Applying UML and Patterns	H37: UML Deployment & Component Diagrams (alleen deployment)	Ja
Patterns of Enterprise Application Architecture	H1	Ja
PluralSight (deze cursus gaat uit van C# in de codevoorbeelden maar de principes en patronen zijn universeel toepasbaar).	https://app.pluralsight.com/library/courses/architecting-applications-dotnet/table-of-contents <ul style="list-style-type: none"> • Real-World Architectural Thinking (25 min) • Business Logic Layer (31 min) • Service Layer (21 min) • Presentation Layer (19 min) • Data Access Layer (39 min) 	Nee

Data Source Layer & JDBC

Bron	Hoofdstuk	Toetsstof
Sheet	07. Data Source Layer JEE.pptx	Ja
Patterns of Enterprise Application Architecture	H2	Ja
Patterns of Enterprise Application Architecture	H10	Ja
PluralSight	https://app.pluralsight.com/library/courses/jdbc-java-platform-working-with-databases/table-of-contents <ul style="list-style-type: none"> • Introduction (18 min) • Getting Started with JDBC (28 min) • Performing Basic CRUD Operations Using JDBC (56 min) https://app.pluralsight.com/library/courses/design-patterns-java-behavioral/table-of-contents <ul style="list-style-type: none"> • Chain of Responsibility Pattern, vooral het voorbeeld over Logging (16 min) 	Ja
Oefening	1. JDBC+Logging Oefening	Ja

Domain Layer, Service Layer & REST

Bron	Hoofdstuk	Toetsstof
Sheet	02. Layer pattern.pptx 05. CORS.pptx	Ja
Patterns of Enterprise Application Architecture	H9. H15, H18	Ja
PluralSight	https://app.pluralsight.com/library/courses/restful-services-java-using-jersey/table-of-contents <ul style="list-style-type: none"> • The Fundamentals of RESTful Services in Java (5 min) • RESTful Architecture (10 min) • Using HTTP GET (51 min) • Using HTTP POST (25 min) 	Ja

Dependency Injection met CDI

Bron	Hoofdstuk	Toetsstof

Sheet	08. CDI JEE.pptx	Ja
PluralSight	<p>Inversion of Control</p> <ul style="list-style-type: none">• Dependency Inversion (31 min)• Inversion of Control (33 min)• Dependency Injection (30 min) <p>Test-Drive Development Practices in Java</p> <ul style="list-style-type: none">• Basic features of Mockito (29 min)• Mockito Features Deep-Dive (38 min) <p>Context and Dependencie Injection (CDI 1.1)</p> <ul style="list-style-type: none">• Injection with CDI (38 min)	Ja
Oefening	1. CDI Oefening Java EE	Ja