

Requirements Enterprise Mobile - JAVA

Unit Test alle logica!

Falende testen zijn niet ok.

@Ignored testen zijn niet ok.

Testen met pointless assertions zijn niet ok, bvb

```
assertThat(true).isEqualTo(true);
```

IntegrationTests

- Alle DomainRepositories/DAO's moeten integratietesten hebben.
- Minstens 1 geautomatiseerde Scenario Test die vertrekt vanaf een Controller en de hele achterliggende flow verifieert.

Voorbeeld scenario: een klant bestelt een broodje en ziet zijn geplaatste bestelling.

POST /order > nieuwe Order entity wordt aangemaakt > Order wordt opgeslaan in de DB,

GET /order > pas aangemaakte Order Entity wordt opgezocht uit de DB en teruggegeven als JSON)

Deployment

Op een application server zoals Tomcat, of als standalone executable jar zoals Jetty of Embedded Tomcat (SpringBoot).

Spring

Concepten DI en IOC grondig kennen, kunnen uitleggen en toepassen

Annotations kunnen uitleggen en toepassen

Geavanceerde configuratie kunnen uitleggen en toepassen

Spring Boot applicatie kunnen opzetten en toelichten.

Controllers (REST)

Controllers die zowel GET, POST, PUT en DELETE aantonen en de verschillen tussen deze uitleggen.

Messaging

Gebruik een JMS Queue in jullie applicatie, denk na over een gepaste case.

Zorg dat je pub/sub concept kan implementeren, toelichten en begrijpt.

Als jullie voor andere doeleinden ook een Queue willen gebruiken, go ahead!

Security

Zorg ervoor dat jullie REST controllers op zijn minst met Basic Authentication secured zijn. Dat wil zeggen dat er telkens een Authentication: Basic header mee gestuurd moet worden met jullie HttpRequests met daarin de Base64 encoded string van username:password. Zorg ervoor dat er op zijn minst 1 POST method van een REST controller autorisatie staat waar er slechts 1 enkele user met die specifieke rol toegang toe heeft. Ook ben je in staat om autorisatie toe te passen gebaseerd op rollen. Zorg dat je de volledige configuratie van Spring Security kan implementeren, uitleggen en begrijpt.

AOP

Wat is AOP en wat betekent AOP binnen Spring. Onderzoek dit grondig, het wordt meer toegepast dan je denkt! Pas zelf een Aspect toe op je applicatie. Zorg dat je het Proxy pattern onder de knie hebt.

Spring Data

Bouw je data laag op adhv Spring Data, zorg voor de implementatie met Repositories en leg de werking uit.

Transacties

Waarom gebruiken we transacties en wat betekent propagatie en isolatie? Doe er meer ervaring over op door gepaste literatuur te lezen/bestuderen. Zorg dat je de volledige configuratie van Spring Transactions kan toepassen in je applicatie en uitleggen.

Samengevat in geziene hoofdstukken

1. Core Spring VOLLEDIG !!!
2. Enterprise Spring - INLEIDING, DATABASE TOEGANG, SPRING DATA, TRANSACTIEBEHEER, BEVEILIGING (authorizatie + authenticatie), MESSAGING, REST WEBSERVICES