

# Requirements Enterprise Mobile - JAVA

## Unit Test all the things!

Falende testen zijn niet ok.

@Ignored testen zijn niet ok.

Testen met pointless assertions zijn niet ok, bvb

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

## Deployment

Kies maar: op een ApplicationServer zoals Tomcat, of als standalone executable jar zoals Jetty of Embedded Tomcat (SpringBoot).

## Spring

DI - IOC

Annotations

Geavanceerde configuratie

Spring Boot

## Controllers (REST Resources)

Een paar @Controllers die zowel GET, POST en PUT demonstreren.

Ken de verschillen tussen deze.

IntegrationTests (geen onderdeel van de cursus, dus niet kennen voor examen)

- Alle DomainRepositories/DAO's moeten integratietesten hebben.
- Minstens 1 geautomatiseerde ScenarioTest 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)

## Messaging

Zorg ervoor dat logging in jullie applicatie weggeschreven wordt via een JMS Queue.

Zorg dat je pub/sub concept kan 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 Authorization: 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 uitleggen en begrijpt.

## AOP

Wat is AOP en wat betekent AOP binnen Spring. Pas zelf een Aspect toe op je applicatie. Zorg dat je het Proxy pattern onder de knie hebt.

## Transacties

Waarom gebruiken we transacties en wat betekent propagatie en isolatie? Lees er ook eens een artikel over op internet.

Zorg dat je de volledige configuratie van Spring Transactions kan uitleggen en begrijpt.

## Samengevat in geziene hoofdstukken

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