Flashbook

Rest-specifikation

Spring är ett stort ekosystem av ramverk och vi tyckte att det skulle vara olämpligt att importera endast delar från detta för att sedan blanda in andra ramverk. Vi bestämde oss istället för att använda Jax-RS som även är Javas default specifikation.

Rest-implementation

Vi valde Jax-rs-implementationen Jersey. Jersey är en referensimplementation från Oracle och vi tänkte att den skulle kunna vara lätt att implementera.

Lastbalanserare

Med godkännande av Reine använde vi Nginx istället för Apache eftersom det blev problem med konfiguration och vi inte ville lägga för mycket tid när lastbalanseraren har en så pass liten roll i arbetet.

Vi fick aldrig lastbalanseraren att fungera pågrund av att adressen som ska hashas till StickySession alltid blev dockers interna address och inte klientens. Vår Nginx server agerar därför i nuläget endast Reverse Proxy och redirectar alla HTTP till HTTPS där vi har ett självsignerat certifikat.

Vi använder en Servlet 3 som låter oss slippa en web.xml i backend. Eftersom xml är depricated så är detta något vi ville göra.

Depdency Manager

Vi använder Maven för att hantera våra depencies och för att kompilera projekten. Detta för att bli oberoende från plattform och utvecklingsmiljö. Vi valde Maven över Gradle trots att Gradle verkar vara snabbare då vi redan har viss förkunskap om Maven samtidigt som vi känner att det inte finns tid att lära sig en ny hanterare.

Databas

Vi valde att använda en MySQL-databas eftersom vi känner oss mest bekväma med relationsdatabaser och har mest erfarenhet av just MySQL.

Docker

Separata containrar för lastbalanseraren, Tomcat-frontend, Tomcat-backend och MySQL-databasen.

Unit Testing

Vi valde det populära ramverket junit och mockito och vi testar Beans, Services och Databasklasser. Getters och setters testas inte eftersom JVM redan testar dessa och det är slöseri med tid för oss att testa dessa.

Authentisering

Authentisering sker med hjälp av JWT.

JSON parser

Jackson användes som JSON parser för att göra om POJO’s till JSON.

JSF och AJAX

JSF används enligt specifikation. Används görs även den extra komponenten PrimeFace för att få en bättre ajax-implementation. Tanken var även att PrimeFaces skulle användas för att kunna skicka med bilder i meddelanden men detta han vi ej med.

ORM mappare

JPA användes som ORM mappare och denna kräver en ”provider” som enligt specifikationen för arbetet ska vara just Hibernate.

ORM provider

Hibernate, men kod är skrivet i JPA så man kan ganska lätt kan byta ut implementationen om man byter ut våra HQL hibernate queries.

Dependency injections

Dependency injections används för att få lägre koppling mellan klasser och för att kunna skapa mock-objekt till testing.