

Mjukvarutestning - inlämningsuppgift

Bakgrund

Mickes yngsta dotter lyckades klicka sig ut från YouTube-kids och in på TUI bland dyra spanienresor...

...någon behöver nu täcka upp för honom så att han kan åka på solsemester.

Detta innebär, tyvärr, för din del att du måste täcka upp för honom då han har en deadline att förhålla sig till.

På hans skrivbord ligger ett Spring Boot-projekt som du behöver skriva tester för och det måste vara klart senast 20 augusti klockan 17.00.

Teknik

Använd dig av funktionalitet från ramverken junit.jupiter, assertJ samt mockito.

Projektet använder sig av en MySQL-databas (i utvecklingsmiljön), men du kommer att använda dig av en H2-databas i testmiljön (gäller enbart för testning av repository).

Utgångsläge

Du får länk till ett repo, ett REST API-projekt. Micke har hunnit påbörja vissa tester men han ber dig fokusera på att skriva tester för repository och servicelager, (StudentRepository, SchoolService & StudentService)

I controllers används MockMvc, detta är alltså inget du ska lägga någon energi på utan det fixar Micke när han kommer tillbaka från soliga Spanien. Absolut kan man se det som ett material att gå till för mer utmaning när man klarat av det andra (Ej betygsgrundande).

Förberedelser

Studera application.properties i projektet och skapa en databas med MySQL Workbench med det namnet som databasen ska ha.

Starta upp applikationen som vanligt, kontrollera att det skapas en tabell i databasen.

Uppgift

Se video med instruktioner för att kunna skriva testfall för ett repository, "KomIgångMedInlämning".

Börja med att skriva testerna för StudentRepository, vilket är den enda klassen vi inte mockar andra klasser i eftersom vi faktiskt vill använda oss av en databas. Fortsätt sedan med StudentService och SchoolService i valfri ordning. Nu får ni tänka på att övriga klasser än just den ni testar behövas mockas.

Här följer en lista med användbara metoder.

assertEquals(), assertNotEquals(), assertTrue(), assertFalse(), assertThat(),
assertThatThrownBy(), assertThrows(), assertDoesNotThrow()

mock()

given(MOCK.METHOD()).willReturn(SOMETHING)
when(MOCK.METHOD()).thenReturn(SOMETHING)
doReturn(SOMETHING).when(MOCK.METHOD())

verify(MOCK, never()).METHOD()
verify(MOCK).METHOD()

För att nå betyget G

75-89% av koden(lines) har coverage och passerar i tester

För att nå betyget VG

90+% av koden(lines) har coverage och passerar i tester

Dessa procent gäller för vardera testklass.