

REST. Test af din funktionalitet dvs. din Manager-klasse

Tidligere opgaver

- [Design REST API](#)
- [Simpel Rest Service](#)
- [Rest med Cors](#)

Du skal i denne opgave lave **testen før** du koder din manager klasse

Opgave 1: Test Driven Development

Opgave 1.1: Model-klasse

Du skal i dit Library-projekt implementere endnu en af dine model klasser (fra din domain model)

Opgave 1.2: Definer et interface til din xxxxManager

Til din nye model-klasse skal du definere (kode) et interface 'IxxxxManager' til din kommende manager-klasse.

Dette interface skal have 4 metoder (en til hvert CRUD), Desuden skal interfacet have en Read til fx et ID, samt endnu en Read til søgning efter fx et navn/model/type.... DVS der skal være mindst 6 metoder defineret.

Du skal så absolut sparsomt implementere din xxxManager-klasse, dvs. hver metode kaster en NotImplementedException

Opgave 1.3 Definer dine test

Du skal for hver metode i dit interface definere, hvilke input du skal teste med, samt hvilke værdier du forventer de returnere – også i fejlsituationer.

Opgave 1.4 Implementere en UnitTest, der dækker alle dine definerede test

Du skal lave et Test-projekt, hvor du kan køre dine test. Dette kan lettest gøres ved du lader 'musen' være over navnet på din klasse, hvorefter du højre-klikker og vælger 'Create Unit Tests' og svarer ok.

Du har nu en unit-testklasse hvor du kan implementere dine test.

(du kan evt. gøre brug af [DataRow(..)], samt [TestInitializer])

Husk at opbygge hver test efter **arrange-act-assert** princippet.

Kør testen -> og den vil blive rød.

Opgave 1.5 implementer metoderne i din xxxManager klasse

Du skal implementere dine 6 – metoder i din manager-klasse – **En ad gangen**

Og for hver gang køre din test som langsomt bliver mere og mere **grøn**, indtil alle metoderne er implementeret og hele testen derved bliver grøn.