# Bibliotheek.Domain

Hier de code map in kopiëren

Bibliotheek.Abstract

Is een aparte namespace om gebruikt te kunnen worden door verchillende namespaces. Bevat IGebruiker. Bedoeling is van een implementatie van een gebruiker te kunnen meegeven aan een uitlening.

Bibliotheek.Domain.Abstract

Bevat IBibliotheekRepository die IDisposable implementeerd (wordt gedisposed in de controller) en de basis legt voor repositories.

Bibliotheek.Domain.Concrete

Bevat EFBibliotheekRepository die de link naar de db voorziet.

* implementeert DbContext en IBibliotheekRepository
* initialiseert de context met de juiste connectionstring (komende van de aanroepende applicatie, hier in de root web.config van de mvc app, in ontwikkeling gebruik ik localdb)
* bevat de nodige DbSets en methods (->interface)

Migrations

Om de code en de database in sync te houden maak ik gebruik van Migrations.

* In de NuGet Package Manager Console:
  + Enable-Migrations *projectname*, projectname kan weggelaten worden als de Default project juist is (bovenaan in de PM)
* Als er nu iets in de code (entities) gewijzigd wordt, een property bij, een annotation, etc, dan kunnen we die met de db syncen met de volgende commandos:
  + Add-Migrations *migrationName,* de naam die we willen geven aan het “scriptje” die de update zal doen ( cs-file te vinden in de map Migrations van het project, die file kunnen we trouwens editen…)
  + Update-Database (evtl met –Verbose om de updates te zien)

Database

In ontwikkeling localdb en in production SQLServer (2012?).

