# Упражнения: ASP.NET MVC като потребителски интерфейс

## Създаване на просто приложение

В рамките на това упражнение ще направим трислойно приложение, като слоевете за данни и услуги ще са аналогични с досега направените, но за презентационен слой ще използваме ASP.NET MVC.

## Структура на проекта

Започнете със създаване на **ASP.NET Web Application** проект **ProductApp**. Използвайте **Visual C#** -> **Web** -> **ASP.NET Web Application**



Сега създайте нов проект, като тук изберете **Class Library.** Това ще бъде проекта за слоя данни **Data.**

****

По сходен начин ще създадем и проект **Business.** Използваме **Class Library**, тъй като във всеки от тези проекти създаваме класове, които да могат да бъдат достъпвани от други проекти.

## Добавяне на EntityFramework и референции

След като сме създали всички проекти е време да добавим **EntityFramework.** Цъкнете с десен бутон върху solution-a и изберете **Manage NuGet Packages for Solution** и инсталирайте **EntityFramework**. Изберете инсталация за **всичките** проекти в solution-a.



Освен това добавете следните референции (десен бутон върху проекта -> Add -> Reference) за всеки от проектите:

* На проект **Business** добавете референция към **Data**
* На проект **ProductApp** /ASP.NET MVC проекта/ добавете референция към **Business** и **Data**

## База данни

Тук ще използвамевече съществуващата от предните проекти **ProductDb**. Отново единственият ни ангажимент е да добавим низа за връзка. В **Web.config** на проект **ProductApp** открийте **<connectionStrings>** тага и в него добавете **<add>** таг с низа за връзката:
****

## Слой за данни

**Слоят за данни тук е напълно аналогичен с този в предното упражнение.**
Тук той се състои от папка с **модел** и **контекст**.

Моделът тук е клас **Product.cs,** находящ се в **Model** папката,като в него описваме единствено свойствата му, които всъщност съответстват на колоните от таблицата.



В самата папка **Data** се намира и **ProductContext** класа. Той трябва да наследява **DbContext**. Тук ще се наложи да добавите и **using** директива, понеже класа **DbContext** e част от **EntityFramework**.
Самият клас ще съдържа конструктор, в който ще има обръщение към конструктора на базовия клас, където за параметър се подава името (**name** атрибутът) от низа за връзка, който добавихме в **App.config** по-рано.

Освен това в конструктора ще имаме и свойство, което ще е от **DbSet<Product>.**

Кодът е както следва:


С това сме готови с нашия слой за данни.

## Бизнес слой

**Слоят за бизнес логиката тук е напълно аналогичен с този в предното упражнение.**

Тук ще имаме поле от тип **ProductContext**, което ще използваме в методите.

Методите като логика работят по абсолютно сходен начин с предното упражнение. Класът изглежда по следния начин:


А методите са съответно:






## Презентационен слой

Уверете се, че сте създали успешно слоя за данни и бизнес слоя. Изпълнете команда **Build** за целия **solution**.

Ако всичко е наред, можем да пристъпим към реализация на нашия презентационен слой. За тази цел ще ползваме уеб интерфейс. Тук няма да се впускаме в подробности как да изменяте външния му вид или да направите допълнително модификации по него, а само ще покажем как се създава.
В **ProductApp** открийте папката **Controllers**. Чрез десен бутон изберете **Add** -> **Controller**. От опциите изберете **MVC Controller with views, using Entity Framework**.

След това трябва да зададете име на вашия контролер, а освен това трябва да зададете и клас на модела, както и клас на контекста:


Натиснете **Add** и оставете на Visual Studio само да свърши останалото за вас ☺

## Употреба на приложението

След всичко това е време да видим нашето приложение в действие – за разлика от досегашните ни проекти тук на мястото за стартиране е изписано името на уеб браузър:


Натиснете бутона и стартирайте смело. Ще се отвори браузъра, в който ще фигурира адрес от вида [**http://localhost:2960/**](http://localhost:2960/). **Имайте предвид, че вместо 2960 е възможно при вас номерът на порта да е друг!**

Към този адрес добавете думичката **Product**. Ще ви излезе страничка, в която са показани всички записи от базата данни с възможност за **редакция** /**Edit**/, **подробности** /**Details**/ и **изтриване** /**Delete**/. Това са CRUD операциите и те сега са налични чрез добре изглеждащ уеб интерфейс.

С това приключваме реализацията на презентационния ни слой и съответно това упражнение.

## Министерство на образованието и науката (МОН)

* Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "**Обучение за ИТ кариера**" на МОН за подготовка по професия "Приложен програмист".

* Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под **свободен** **лиценз CC-BY-NC-SA** (Creative Commons Attribution-Non-Commercial-Share-Alike 4.0 International).

