

ASP.NET MVC като потребителски интерфейс



Учителски екип

Обучение за ИТ кариера

<https://it-kariera.mon.bg/e-learning/>



Съдържание

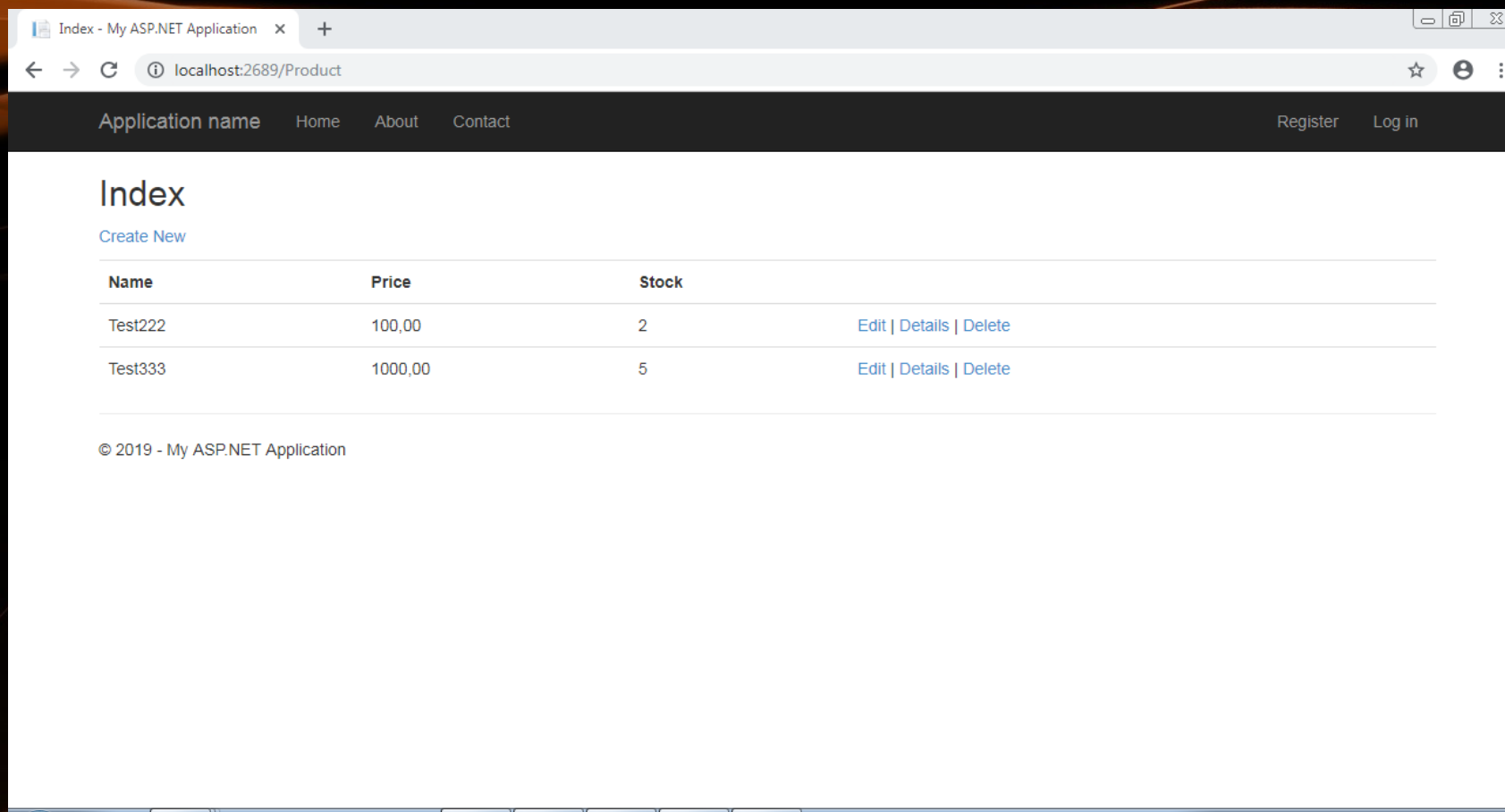
1. Използване на ASP.NET MVC за презентационен слой
2. Разделяне на приложението на проекти според слоя



Какво е ASP.NET MVC

- ASP.NET MVC е рамка за изграждане на уеб приложения.
 - Базирана на MVC шаблона
 - Лесна и бърза за работа
 - Чудесен визуален интерфейс за нашето приложение

Благодарение на ASP.NET MVC ние ще дадем добър и удобен външен вид на нашето CRUD приложение



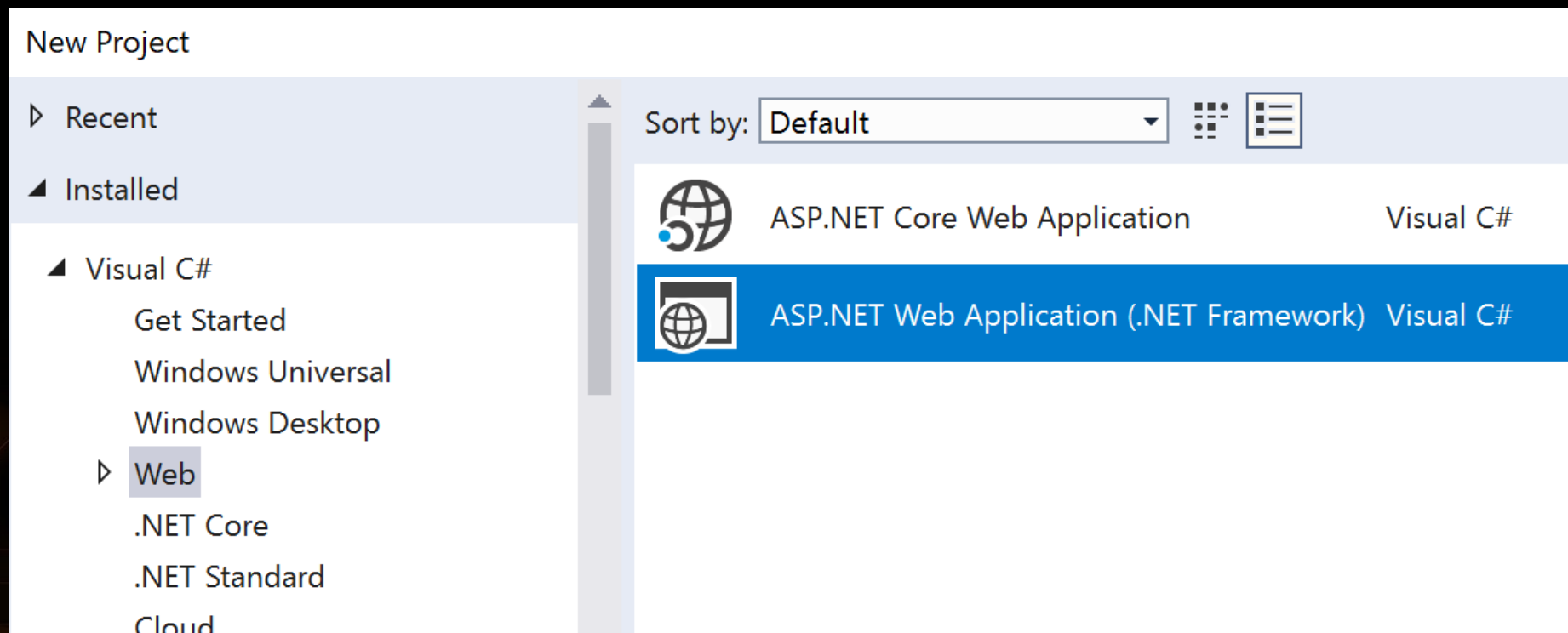
Структура на проекта ни

С участието на ASP.NET MVC

Какви проекти трябва да имаме?

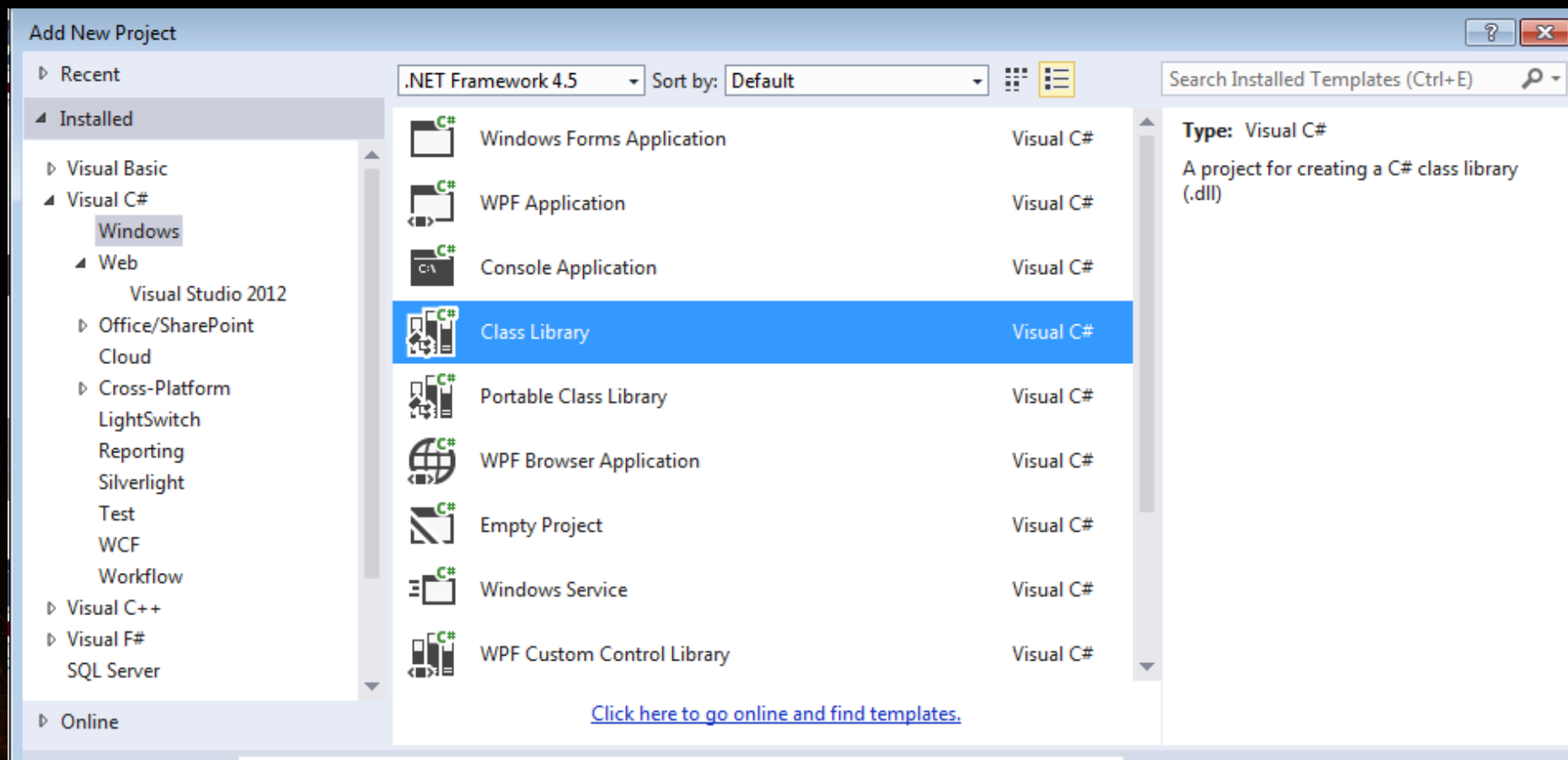
- ASP.NET Web Application

- Това ще е проекта ни, който ще действа като презентационен слой



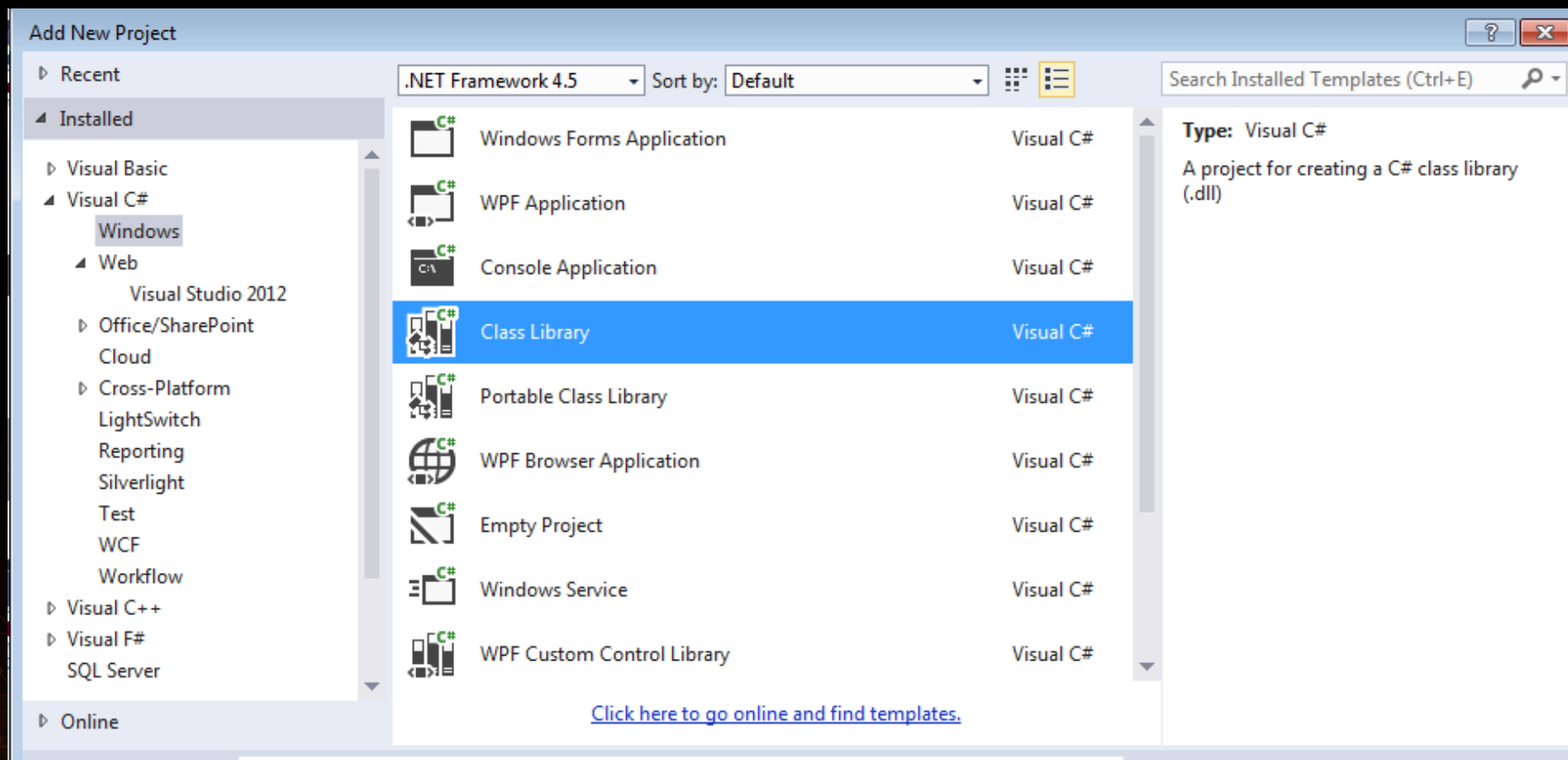
Какви проекти трябва да имаме? (2)

- **Data** - това ще е проекта ни, който ще бъде слоен за данни. При създаването му от Visual C# -> Windows трябва да изберем **Class library**



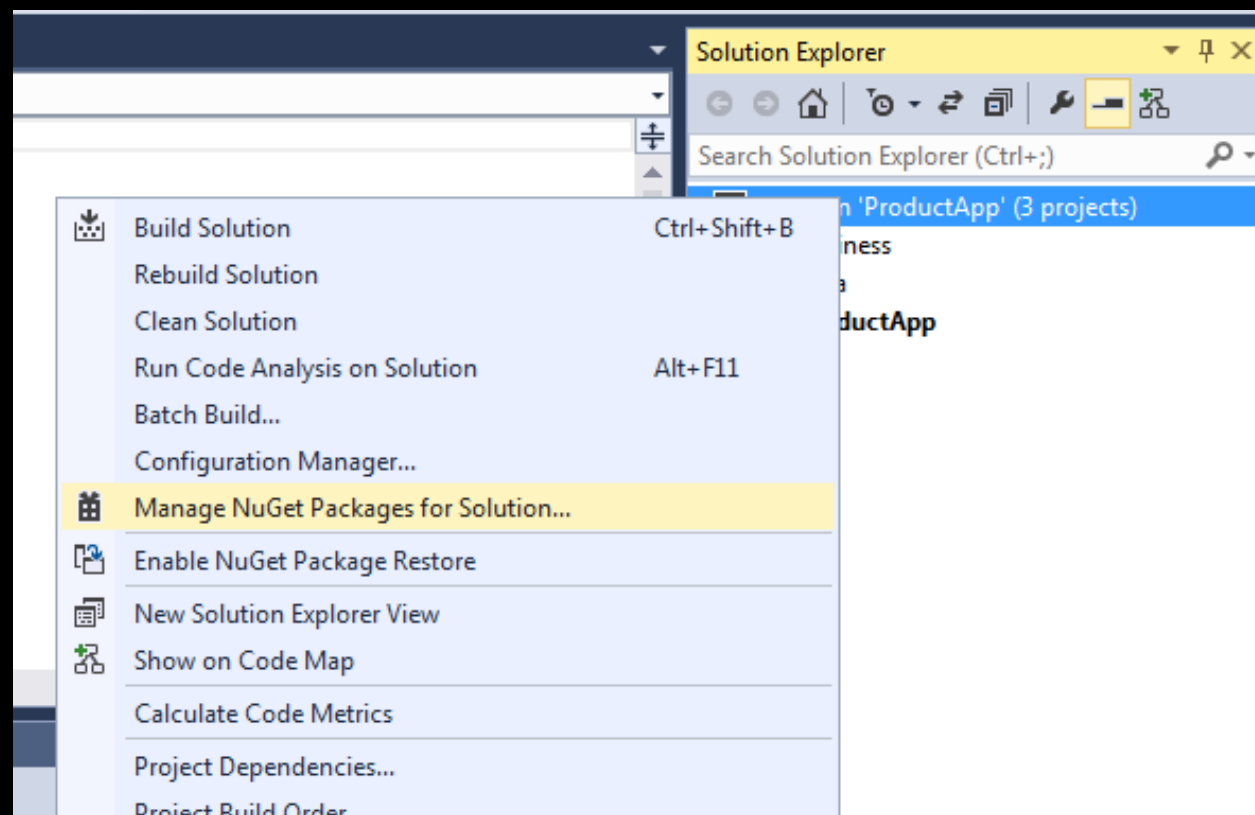
Какви проекти трябва да имаме? (3)

- **Business** - това ще е проекта ни, който ще бъде слоен за услуги. При създаването му от Visual C# -> Windows трябва да изберем **Class library**



Добавяне на EntityFramework

- Цъкнете с десен бутон върху solution-а и изберете Manage NuGet Packages for Solution и инсталирайте EntityFramework. Изберете инсталация за всичките проекти в solution-а



Добавяне на референции

- На проект **Business** изберете [Add]->References. Добавете референция към **Data**
- На проект **ProductApp** /проектът на ASP.NET/ изберете [Add] -> References. Добавете референция към **Business** и **Data**

Data/Model/Product.cs

- Създайте подпапка Model в Data, а в нея модел клас **Product.cs**

```
public class Product
{
    public int Id { get; set; }
    public string Name { get; set; }
    public decimal Price { get; set; }
    public int Stock { get; set; }
}
```

Data/ProductContext.cs

- Създайте **ProductContext** клас, който наследява **DbContext**. Ще има нужда да добавите и **using** директива:

```
using System.Data.Entity;

public class ProductContext : DbContext
{
    public ProductContext()
        : base("name=ProductContext")
    {
    }
    public DbSet<Product> Products { get; set; }
}
```

Business/ProductBusiness.cs

- Създаваме управляващо поле от клас **ProductContext**:

```
private ProductContext context;
```

- Създаваме методи, които извършват желаните операции, извиквайки свойството **Products** и методи върху него, познати ни от **LINQ**:

```
public List<Product> GetAll(){  
    using (productContext = new ProductContext())  
    {  
        return productContext.Products.ToList();  
    }  
}
```

- По аналогичен начин реализираме другите методи

Свързване на ASP.NET с другите слоеве

- В `Web.config` намираме `<connectionStrings>` секцията. Там под другия `<add>` добавяме и нашия:

```
<add name="ProductContext" connectionString="Data Source=.; Initial Catalog=ProductDb; Integrated Security=true,, providerName="System.Data.SqlClient" />
```

- След тази стъпка трябва да се уверим, че сме направили **Build** на целия **Solution**

Свързване на ASP.NET с другите слоеве (2)

- Добавяме контролер, чрез десен бутон върху Controllers папката -> Add -> Controller
- Изберете MVC Controller with views
- За име на контролера задайте ProductController
- За Model клас посочете Product (Data.Model)
- За Data Context клас посочете ProductContext (Data)

Използване на приложението

- Компилирайте и стартирайте приложението.
- Добавете /Product в края на URL адреса и заредете страницата
- Пред вас вече е наличен уеб интерфейса за нашето приложение, в който може да извършите всички CRUD операции

Използване на приложението (2)

Index - My ASP.NET Application

localhost:2689/Product

Application name Home About Contact Register Log in

Index

[Create New](#)

Name	Price	Stock	
Test222	100,00	2	Edit Details Delete
Test333	1000,00	5	Edit Details Delete

© 2019 - My ASP.NET Application

ASP.NET MVC като потребителски интерфейс



Въпроси?

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"



Министерство
на образованието
и науката



Национална
програма
„Обучение за
ИТ кариера“

- Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под свободен лиценз **CC-BY-NC-SA**



SoftUni
Foundation

