# Демо проект Dummy Shop

Проект для демонстрации совместного использования нескольких технологий .Net в одном приложении.

Для работы вам необходимо развернуть локально базу данных PRODUCTS из бекапа PRODUCTS.bak. А также проверить, что локально у вас есть тестовые данные по адресу .\src\DummyShop\Tests\TestData.

## Технологии

### WPF + MVVM

Используется для реализации графического пользовательского интерфейса.

Паттерн MVVM используется для демонстрации возможности отвязаться от сервиса и работать с тестовыми данными, расположенными в локальной директории

### Ado.Net, SQL

Работа с базой данных

### Web API 2.0

Сервис, хостинг в консольном приложении и авто сгенерированные веб-странички с описанием интерфейсов сервиса

### Microsoft.VisualStudio.TestPlatform

Библиотека для создания и запуска юнит-тестов

### log4net

Библиотека для работы с лог файлами

## Основные части проекта

Основные компоненты расположены в соответствующих каталогах (директориях) проекта

### DAL (data access level)

#### DTO

Проект содержит все объекты DTO (data transfer object), которые используются для работы с данными, возвращаемыми со стороны MS SQL Server

#### DataStorage

Запускает SQL запросы и оборачивает их результаты в объекты DTO

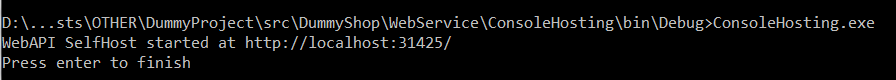
Ключ connectionString в файле App.config определяет строку подключения к базе.

За запуск методов отвечают классы XXXDataProvider, обертка результатов sql запроса в DTO объекты происходит в классах XXXDTOMapper

### WebService

ProductsService – проект с реализацией веб-сервиса. Код для работы с help-страничками расположен тут: .\DummyShop\WebService\ProductsService\Areas\HelpPage\

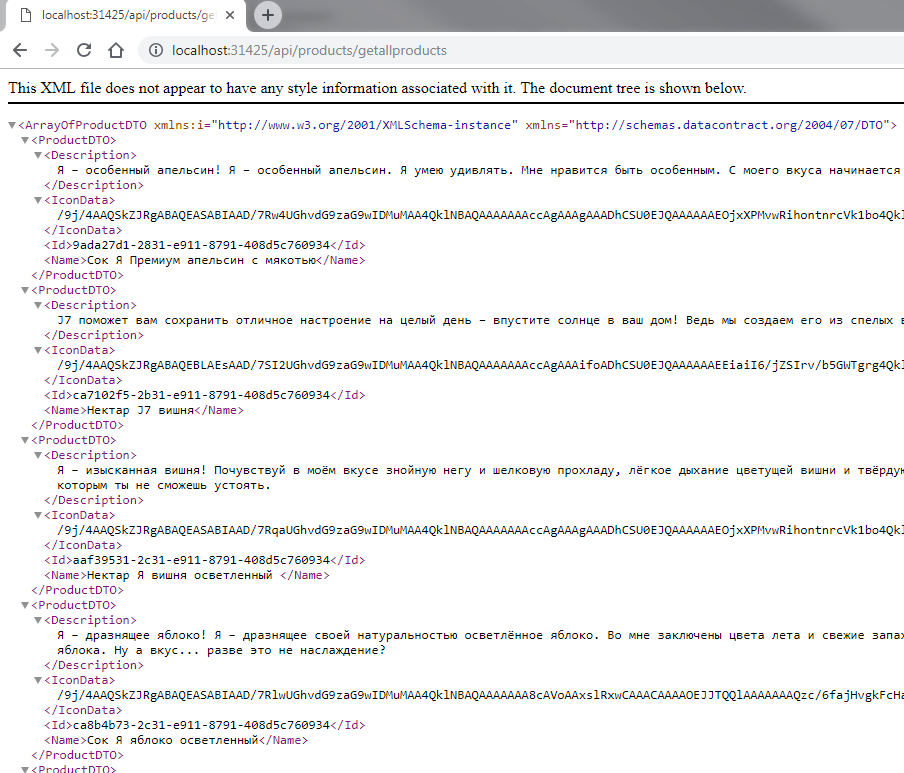
ConsoleHosting – проект для хостинга сервиса в консольном приложении. Для работы с сервисом должно быть запущено приложение ConsoleHosting.exe:



Протестировать сервис вы можете из браузера, запустив, например, url:

http://localhost:31425/api/products/getallproducts

В случае успеха вы увидите что-то такое:



### UI

Использован паттерн MVVM:

Model – View (файл MainWindow.xaml) – ViewModel

Секция View (UI) работает с данными через DataContext, который определяется во ViewModel (ApplicationViewModel).

Для работы с сервисом ViewModel должен использовать класс ProductSupplier, т.е. в методе MainWindow() (файл MainWindow.xaml.cs) DataContext должен быть определен как:

DataContext = new ApplicationViewModel(ProductSupplier.Create());

Для работы с тестовыми данными ViewModel должен использовать класс ProductSupplierMock:

DataContext = new ApplicationViewModel(ProductSupplierMock.Create());

Логирование реализовано через библиотеку log4net и файл .\DummyShop\UI\ClientApp\Other\Logger.cs

Настройки логгера указаны в секции log4net в файле .\DummyShop\UI\ClientApp\App.config

### Tests

Содержит проекты для тестирования бизнес-логики для работы с базой данных (DataStorage.Test) и тестирования работы веб-сервиса (ProductService.Test).