## Среда разработки

Windows 10 Home

MS SQL Server 2017

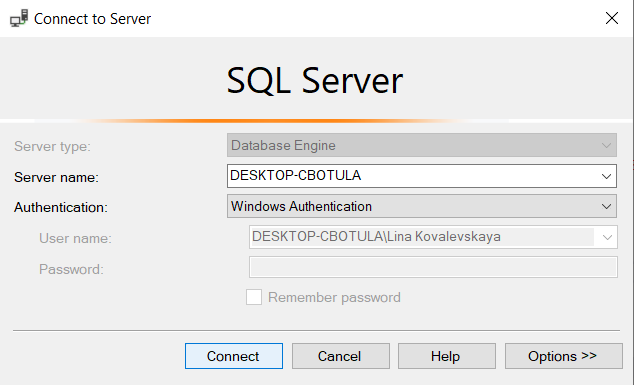
Microsoft SQL Server Management Studio 2018

Visual Studio Community 2019

Dot net cli 3.0.101

## Создание базы данных

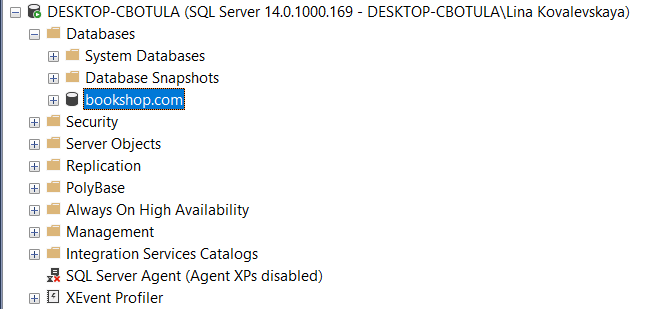
1. Законнектиться как системный юзер



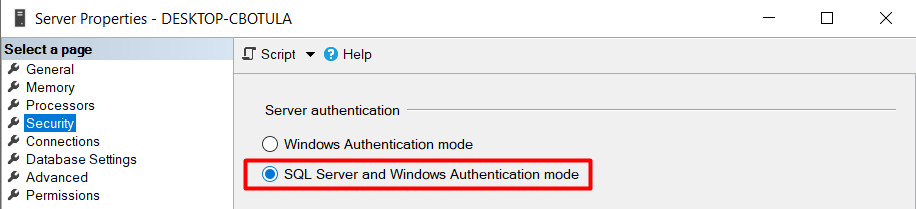
1. Создать бд bookshop.com



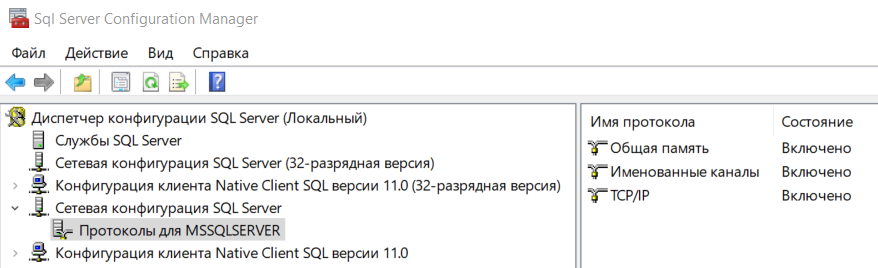
Созданная бд



1. Создать пользователя для работы с бд
   1. Проверить, что в настройках сервера выбрано SQL Server and Windows Authentication

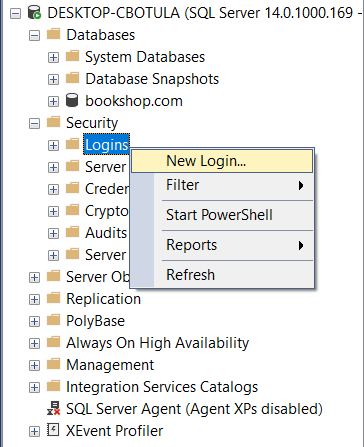


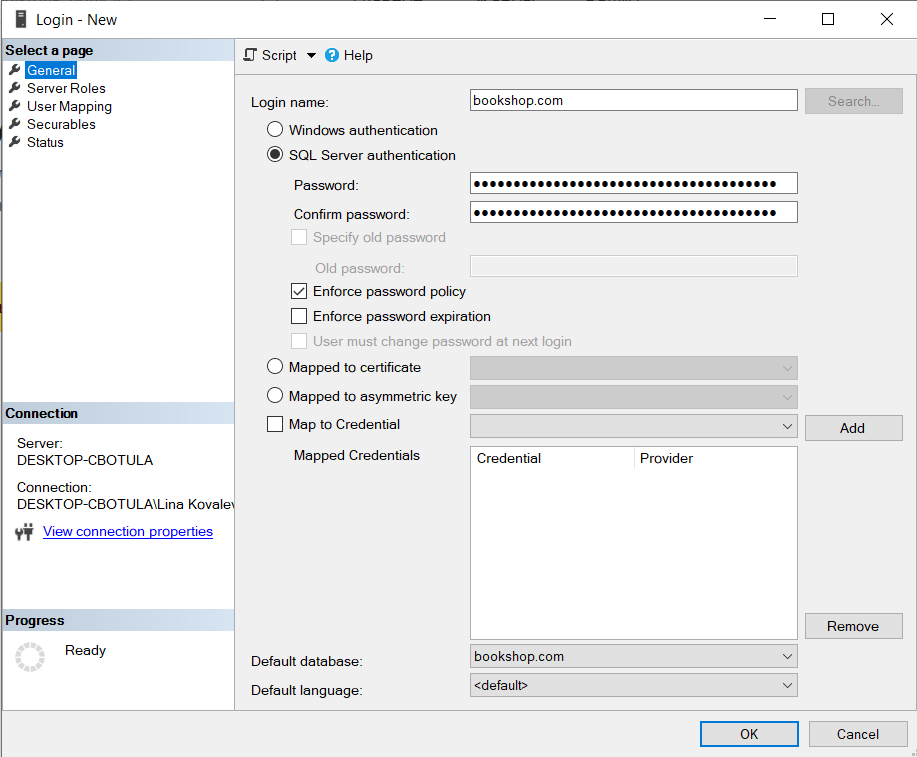
* 1. Проверить сетевую конфигурацию сервера
     1. Запустить диспетчер конфигурации SQL Server 2017
     2. Убедиться, что все 3 параметра (общая память, именованные каналы, TCP/IP) – включено

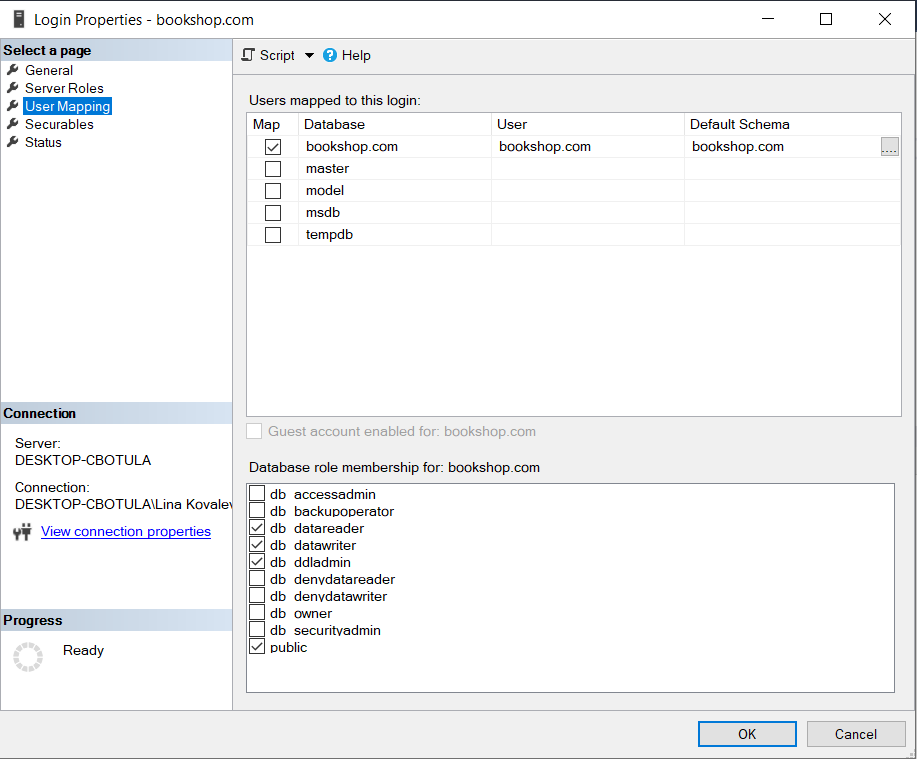


* + 1. Создать учетную запись на сервере

bookshop.com {0BA77AF7-8FEA-48FB-9FD3-9886D343DAAE}







Укажем роли для бд

<https://docs.microsoft.com/en-us/sql/relational-databases/security/authentication-access/database-level-roles?view=sql-server-2017>  
– db datareader

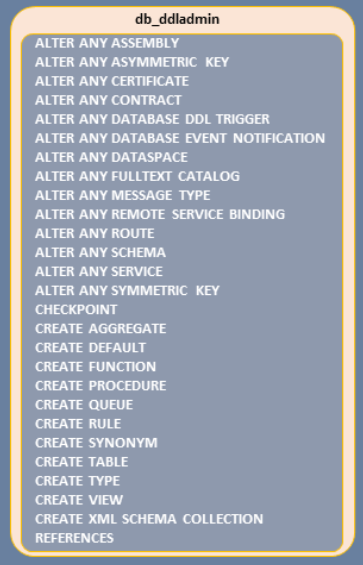
Members of the db\_datareader fixed database role can read all data from all user tables. (grant select on database)

* db datawriter (grant insert, update, delete)

Members of the db\_datawriter fixed database role can add, delete, or change data in all user tables.

* db ddladmin

Members of the db\_ddladmin fixed database role can run any Data Definition Language (DDL) command in a database.



* public (по умолчанию всегда выбрана)

<https://www.sqlnethub.com/blog/the-public-database-role-in-sql-server/>

If we check the securableS for the Pulbic database role in AdventureWorks2014 security, we can see that the Public database role is granted with SELECT access to certain system catalog views such as:

sys.all\_columns

sys.all\_objects

sys.all\_parameters

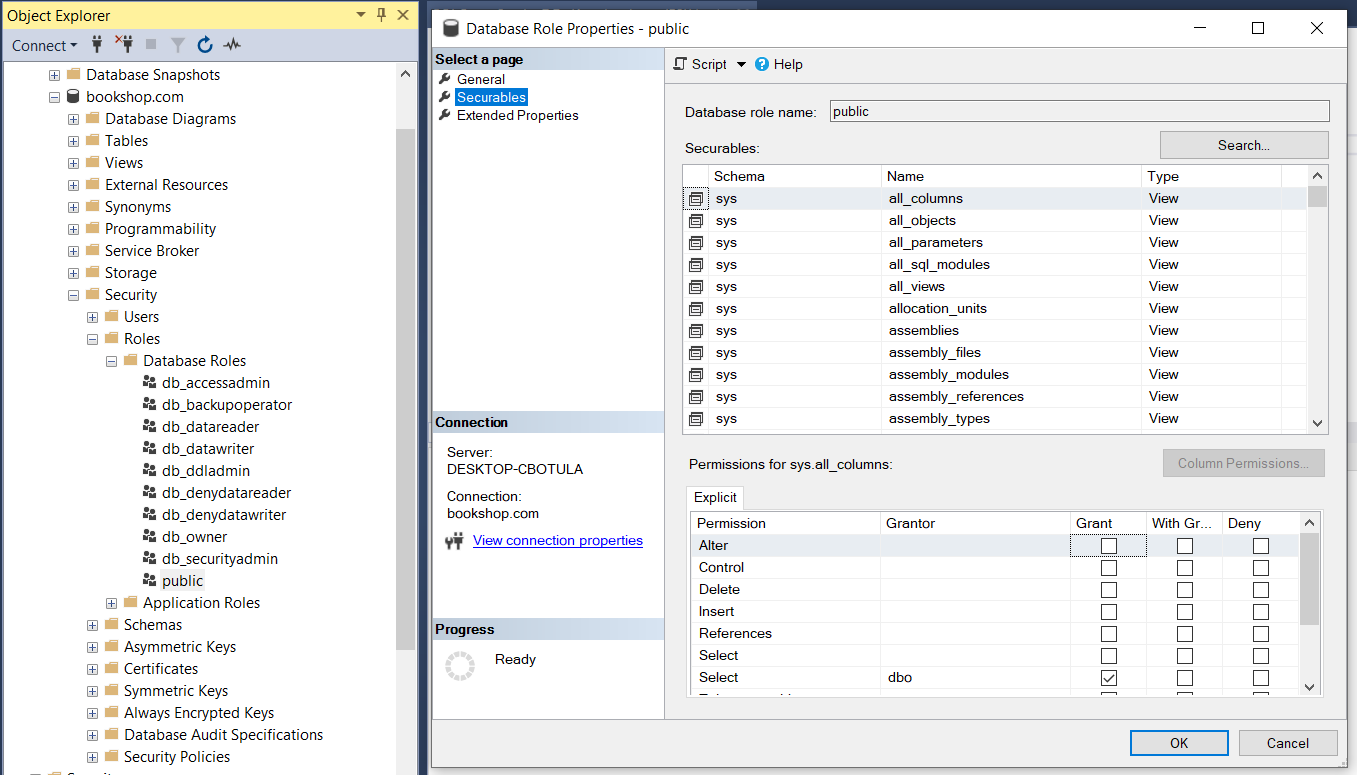
sys.all\_sql\_modules

sys.all\_views

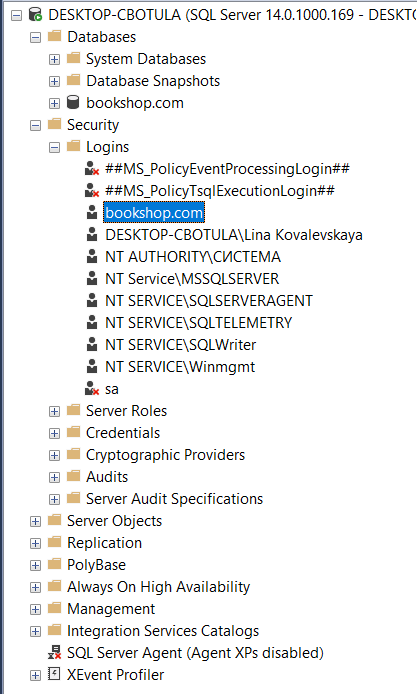
sys.allocation\_units

sys.assemblies

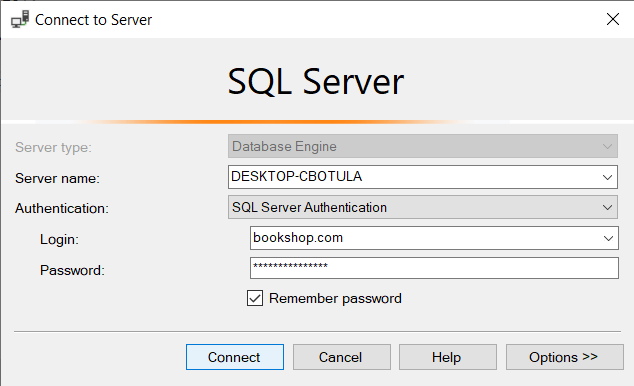
etc.



Созданный пользователь



1. Залогинимся как этот пользователь



В дальнейшем строка подключения выглядит так  
Server=DESKTOP-CBOTULA;Database=bookshop.com;User Id=bookshop.com;Password={0BA77AF7-8FEA-48FB-9FD3-9886D343DAAE};

<https://www.connectionstrings.com/sql-server/>

## Заселение бд

<https://github.com/HeikeMyer/OnlineBookstore/commit/c78685e601411a0dd7d0249bdff85cdaeed84600>

<https://github.com/HeikeMyer/OnlineBookstore/commit/8214d9ca96680ca653670ce2d77b64adabda1d45>

## Создание проекта

Установить dotnet ef

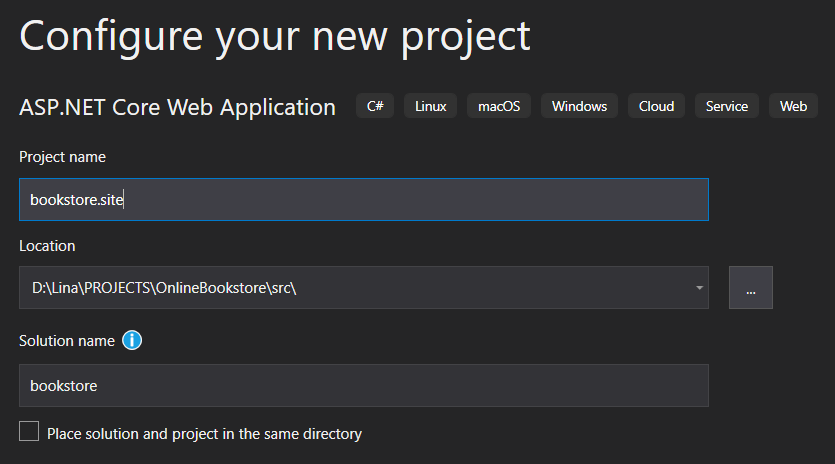
Например, в Package Manager Console  
dotnet tool install --global dotnet-ef --version 3.0.0

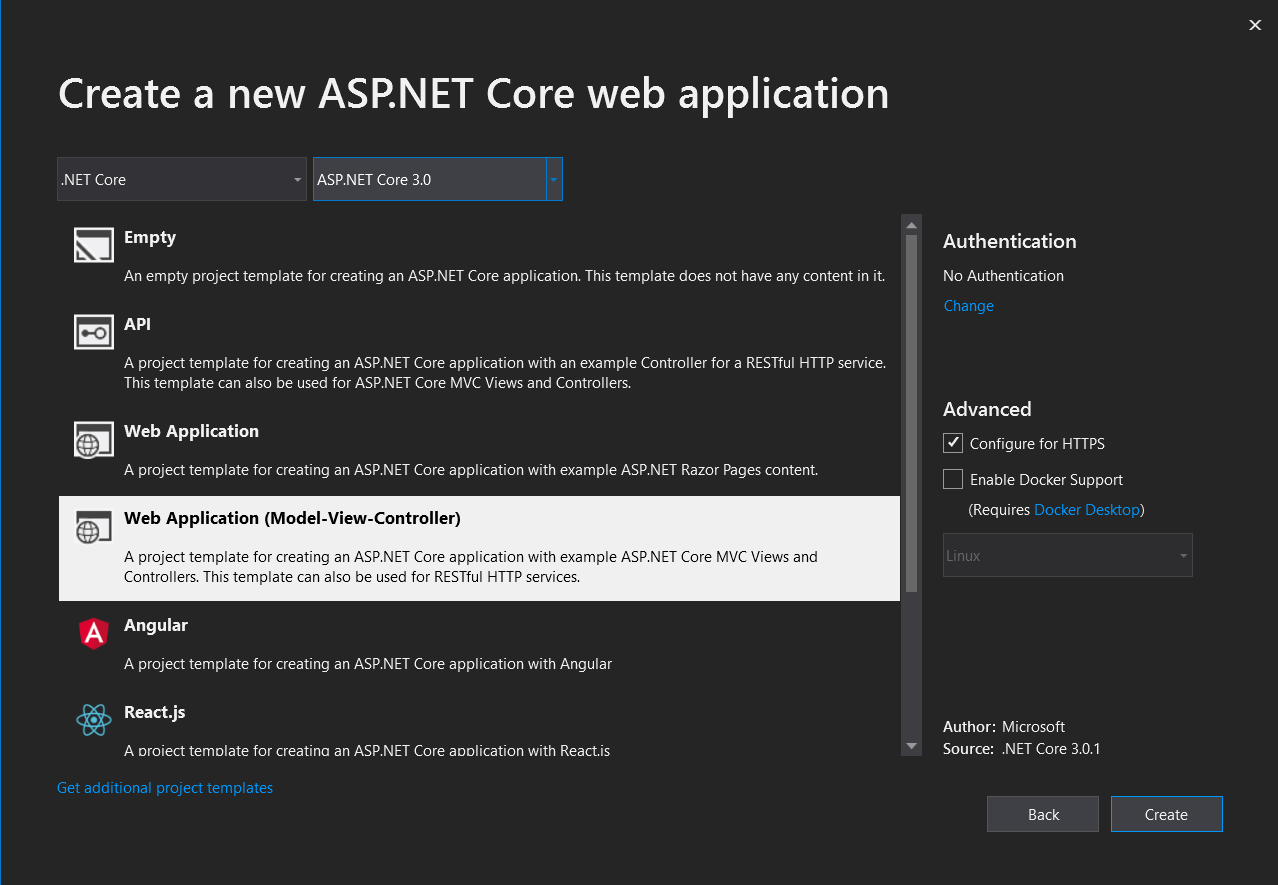
<https://docs.microsoft.com/en-us/ef/core/miscellaneous/cli/dotnet>

<https://github.com/aspnet/EntityFrameworkCore/issues/15448>

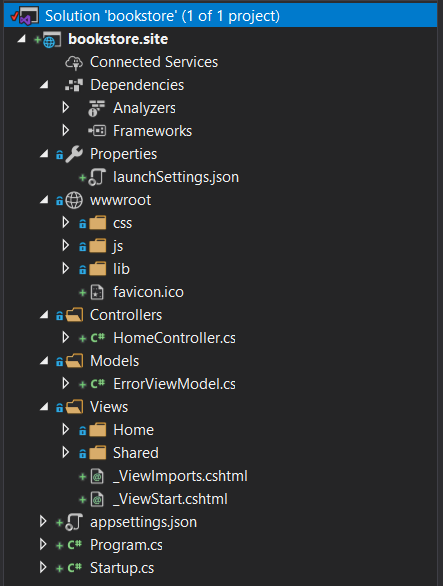
1. Create new project

ASP.NET Core Web Application (ASP.NET Core 3.0)



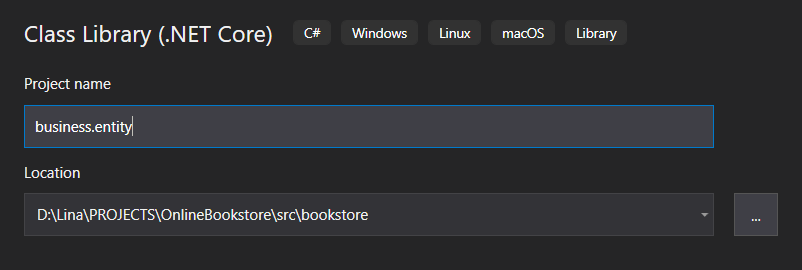


Начальный проект

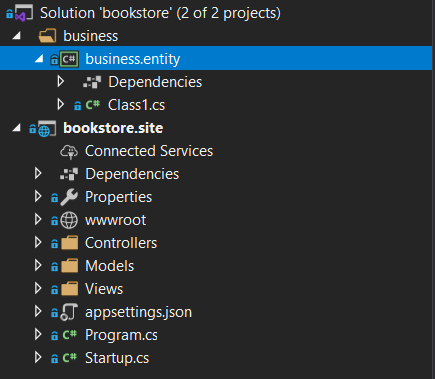


Создадим solution folder business

1. Add project Class Library (.NET Core) в папку business



Созданный проект



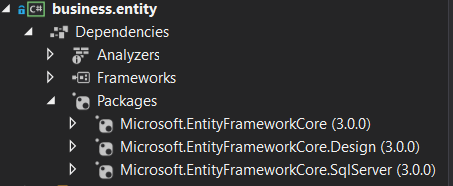
<https://www.learnentityframeworkcore.com/walkthroughs/existing-database>

Добавим NuGet пакеты (3.0.0)

Microsoft.EntityFrameworkCore

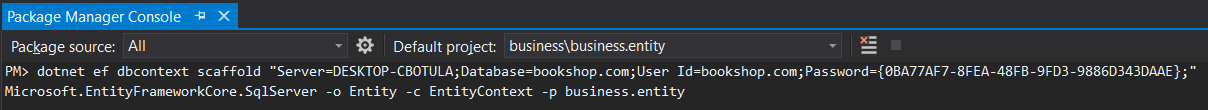
Microsoft.EntityFrameworkCore.SqlServer

Microsoft.EntityFrameworkCore.Design



По существующей базе данных сгенерим классы сущностей

dotnet ef dbcontext scaffold "Server=DESKTOP-CBOTULA;Database=bookshop.com;User Id=bookshop.com;Password={0BA77AF7-8FEA-48FB-9FD3-9886D343DAAE};" Microsoft.EntityFrameworkCore.SqlServer -o Entity -c EntityContext -p business.entity



Получили файлы

