Tallinna Tööstushariduskeskus

Noorem tarkvaraarendaja eriala  
  
  
  
  
  
  
 **VEEBIRAKENDUS “VEEBIPOOD” LOOMINE**   
Projektitöö

**Õpilased:** Denis Saiko  
 **Rühm:** TARpv21  
 **Juhendaja:** IT kutseõpetaja Marina Oleinik

Tallinn 2023

**Sisukord**

[Sissejuhatus 3](#__RefHeading___Toc5628_696080894)

[Seadistamine 4](#__RefHeading___Toc5630_696080894)

[ApplicationDbContext 5](#__RefHeading___Toc5632_696080894)

[Andmebaasi aadress lisamine 6](#__RefHeading___Toc5634_696080894)

[Program.cs fail 7](#__RefHeading___Toc5636_696080894)

[Mudelite lisamine 8](#__RefHeading___Toc5638_696080894)

[Lisame mudelid andmebaasile 10](#__RefHeading___Toc5640_696080894)

[Kontrollerid 11](#__RefHeading___Toc5642_696080894)

[Andmebaasi töö kontroll 13](#__RefHeading___Toc5644_696080894)

[Kasutatud kirjeldus / allikad 16](#__RefHeading___Toc5646_696080894)

[Kokkuvõtte 17](#__RefHeading___Toc5648_696080894)

[Pilt 1. ApplicationDbContext.cs 5](#Pilt!2|sequence)

[Pilt 2: Andmebaasid 6](#Pilt!0|sequence)

[Pilt 3: Program.cs 7](#Pilt!1|sequence)

[Pilt 4. Mudelite kaust 8](#Pilt!4|sequence)

[Pilt 5. Product mudel 8](#Pilt!5|sequence)

[Pilt 6: Product.cs 9](#Pilt!3|sequence)

[Pilt 7: Person.cs 9](#Pilt!6|sequence)

[Pilt 8: Order.cs 9](#Pilt!8|sequence)

[Pilt 9: Category.cs 9](#Pilt!10|sequence)

[Pilt 10: CartProduct.cs 9](#Pilt!11|sequence)

[Pilt 11. Kontrollers kaust 11](#Pilt!7|sequence)

[Pilt 12: Kontrolleri nädis 1 osa 11](#Pilt!12|sequence)

[Pilt 13: Kontrolleri nädis 2 osa 12](#Pilt!13|sequence)

[Pilt 14. Localhost ip 13](#Pilt!9|sequence)

[Pilt 15: Postman Person 13](#Pilt!14|sequence)

[Pilt 16: Postman Category 14](#Pilt!15|sequence)

[Pilt 17: Postman Product 14](#Pilt!16|sequence)

[Pilt 18: Postman CartProduct 15](#Pilt!17|sequence)

[Pilt 19: Postman Order 15](#Pilt!18|sequence)

# Sissejuhatus

Käesoleva töö eesmärgiks on veebipoe teemalise andmebaasi kujundamine ja loomine. Selles rakenduses on mõned kategooriad nagu: Product, Person, Category, Order, CartProduct. Igas nendes kategoorias on võimalus lisada kustutada andmeid.

Programmi loomiseks kasutatakse Microsoft Visual Studio.

Ma kasutan Microsoft Visual Studio, sest see on mugav ja popularseim. Ma töötan juba kolm aastat Visual Studios ja juba on kogemus ja tean mis ja kuidas töötab.

Tulevikus saab seda rakendusi kasutada edasi ja areneda nagu tavaline veebipood.

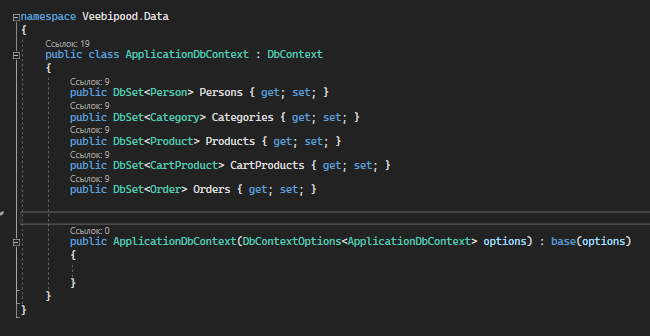
# Seadistamine

Esiteks on vaja installida kolm Framework’i need on: EntityFrameworkCore, EntityFrameworkCore.SqlServer ja EntityFrameworkCore.Tools need kolm peaks olema installitud enne projekti käitlemise.

# ApplicationDbContext

Loome projektis kaust „Data” ja sisse loome veel klassi „ApplicationDbContext” ja kasutame selle klassi nagu andmebaasi konfiguratsiooni fail.

Määrame meie andmebaasi kõik tabelid selles classis:

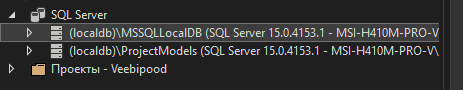


Pilt 1. ApplicationDbContext.cs

# Andmebaasi aadress lisamine

Lisame andmebaasi aadressi appsettings.json faili.

Ülemisest menüüst "View" -> "SQL Server Object Explorer" -> Noolest lahti vajutada SQL Server -> localDB peal parem klõps + rename -> copy

  
Pilt 2: Andmebaasid

Kopeeritud andmetega minge faili appsettings.json ja loome järgmised read:

"ConnectionStrings": {

"DefaultConnection": "server=;database=Veebipood;"

}

Lisame server väljal varem kopeeritud andmed

Database lõppuks kirjutage oma adnmebaasi nimi näiteks „Veebipood”.

Lõppus meie appsettings.json fail on järgnev:

{

"ConnectionStrings": {

"DefaultConnection": "server=(localdb)\\MSSQLLocalDB;database=Pood;"

},

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft.AspNetCore": "Warning"

}

},

"AllowedHosts": "\*"}

# Program.cs fail

Järgmisena avame Program.cs faili lisame sinna järgmised read. Tegemist on äsjaleitud andmebaasi sidumisega (pane tähele, et mõlemas on kirjas "DefaultConnection") ning ApplicationDbContext sidumisega.

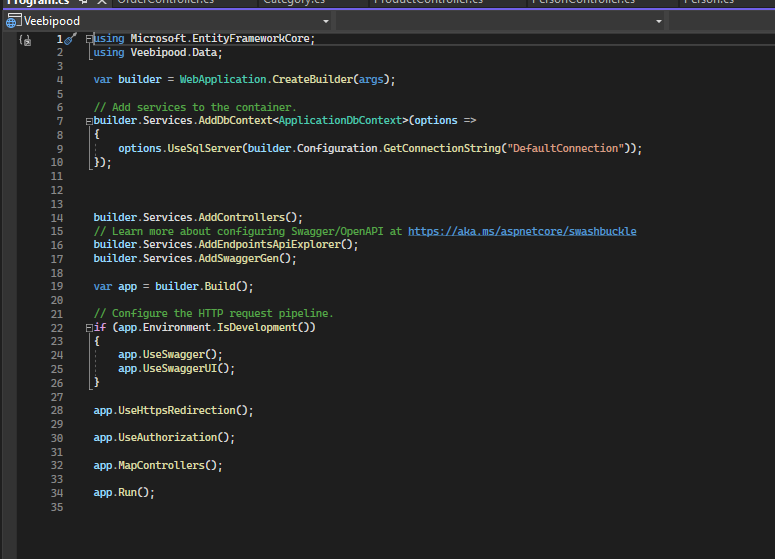
builder.Services.AddDbContext<ApplicationDbContext>(options =>

{

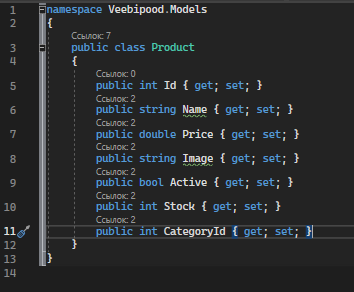
options.UseSqlServer(builder.Configuration.GetConnectionString("DefaultConnection"));

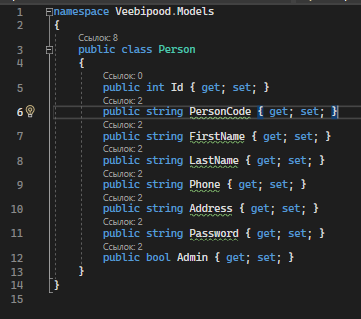
});

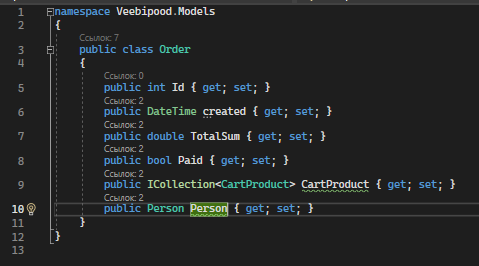
Neid on vaja meile, et seadistada andmebaas. Pärast reade lisast Program.cs fail on järgnev:

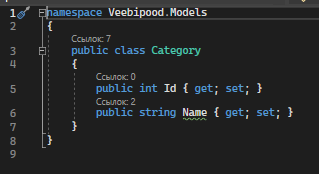
  
Pilt 3: Program.cs

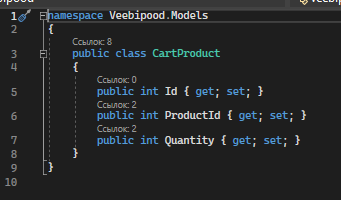
# Mudelite lisamine

  
Pilt 6: Product.cs

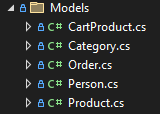
  
Pilt 7: Person.cs

  
Pilt 8: Order.cs

  
Pilt 9: Category.cs

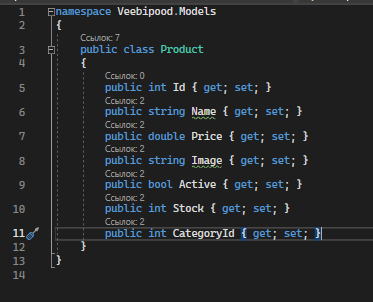
  
Pilt 10: CartProduct.cs

Pärast kõikie enne tehtud etapid lisame kausta „Models” kuhu loome klassid nagu pildis



Pilt 4. Mudelite kaust

Pärast avame klassid ja teeme seal andmebaasi andmemudeli kuju. Näide on pildis allal.



Pilt 5. Product mudel

Üks mudelitest kus ma tegin vormi.

Lõppuks peaks tulla välja nii:

# Lisame mudelid andmebaasile

Edasi on vaja lisada kõik mudelid andmebaasile. See on väja teha iga kord kui loome uue mudeli ja teeme selle mudeli vorm.

Selleks leiame konsool mööda seda teed: "Tools" -> "Nuget Package Manager" -> "Package Manager Console".

Pärast, me kirjutame päringut igale mudelitele.

Päringut mis peavad olema:

1. add-migration addedCartProduct
2. add-migration addedCategory
3. add-migration addedOrder
4. add-migration addedPerson
5. add-migration addedProduct

Pärast kõkide päringute kirjutamise me peame andmebaasi uuendada.

update-database

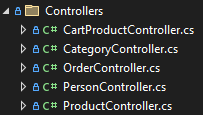
See on päring et uuendada andmebaas. Kui tulevikus teil on vaja teha mingeid redigeerimised mudelites, siis teil on vaja järgmine päring:

"Add-Migration UpdateColumnName"

Pärast iga redigeerimise andmebaasis on vaja kirjutada päringu, et uuendada andmebaasi.

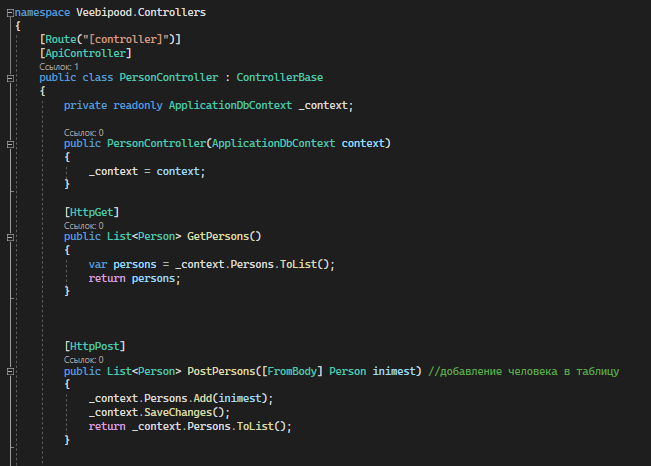
# Kontrollerid

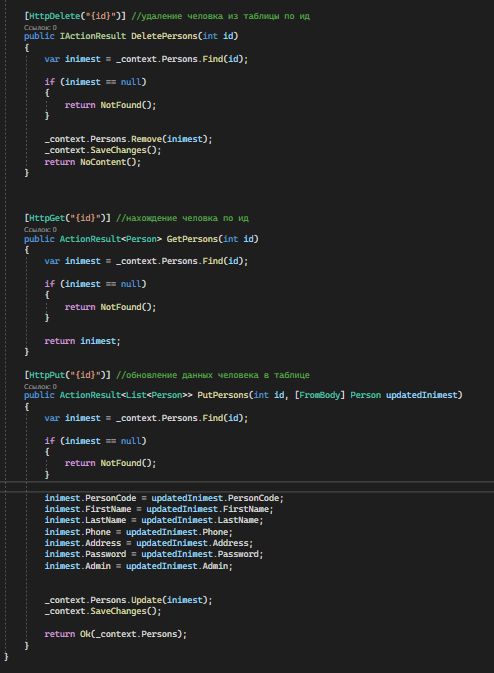
Igale klassidele on vaja oma kontroller, siis lisame neid. Aga enne kontrollide lisamine on vaja lisada ka kaust „Controllers”.   
Controller’i kaust sisu mis peaks olema:

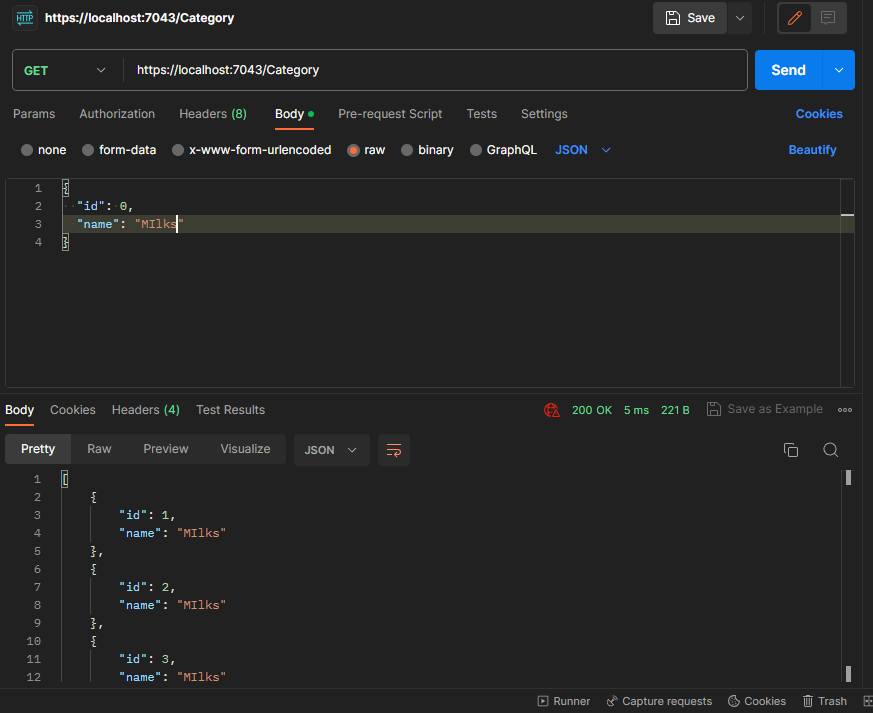


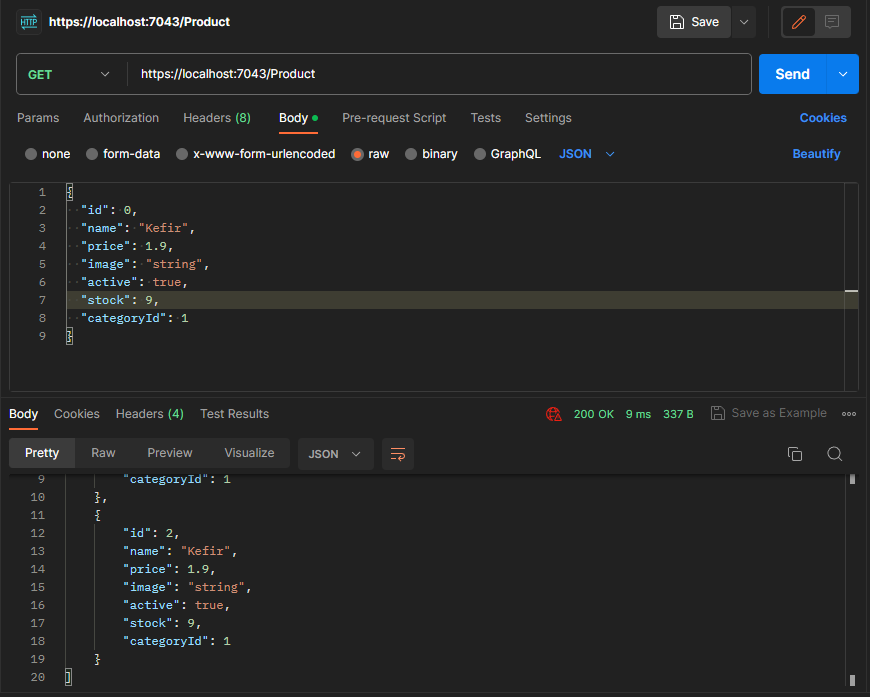
Pilt 11. Kontrollers kaust

Nüüd on vaja konfigureerida need kontrollerid. Sellest oli tehtud järgmised päringud:

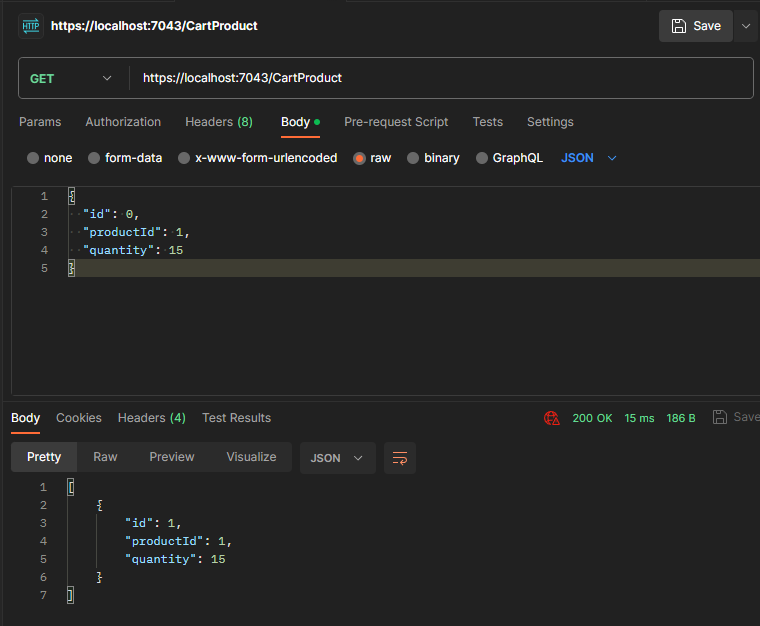
  
Pilt 12: Kontrolleri nädis 1 osa

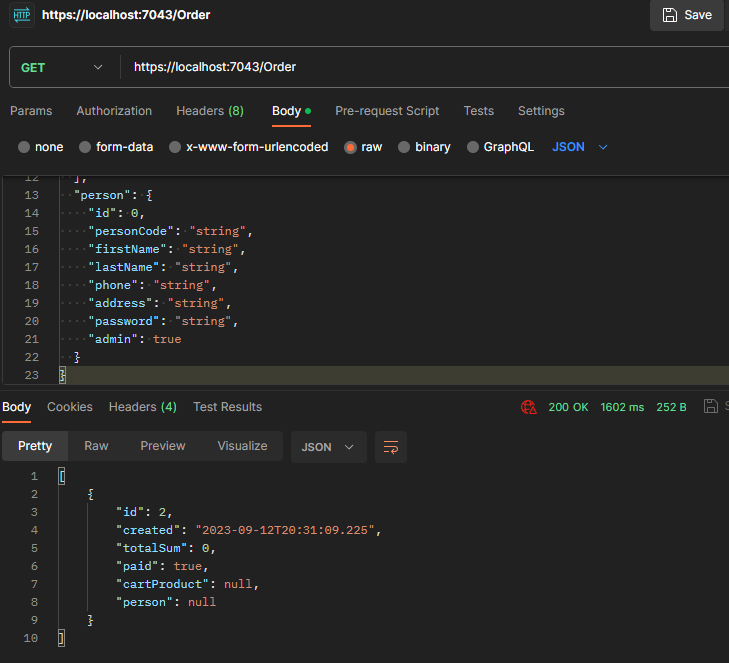
  
Pilt 13: Kontrolleri nädis 2 osa

  
Pilt 16: Postman Category

  
Pilt 17: Postman Product

# Andmebaasi töö kontroll

  
Pilt 18: Postman CartProduct

  
Pilt 19: Postman Order

Andmebaasi kontrollimiseks mina kasutasin rakenduse „Postman”. Kontrollimiseks on vaja leija oma linki, kõige lihtsaim kuidas leia on vaja kävita rakendusi ja siis tuleb ip. Mul on https://localhost:7043/.

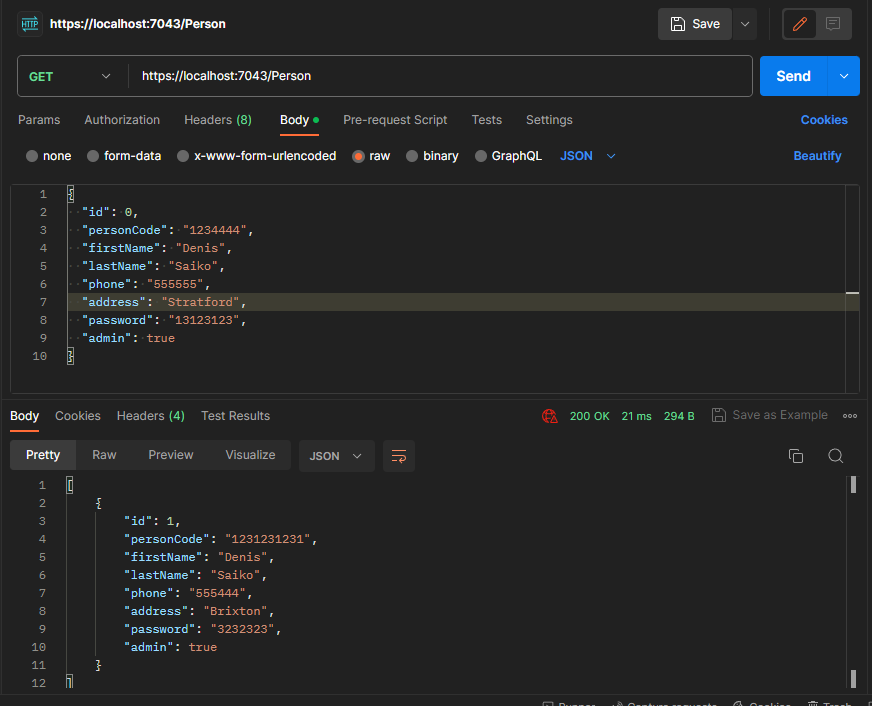


Pilt 14. Localhost ip

Nüüd kui meil on link saame avada Postman’i ja kontrollida päringutega kuidas töötab andmebaas.

Siit ma proovisin teha “Person” päring, ja kõik töötab

Nii saama ma tegin kõigiga päringuga:

  
Pilt 15: Postman Person

Kui kõik need päringud töötavad hästi, siis kõik oli tehtus korrektselt.

# Kasutatud kirjeldus / allikad

1. TalTech juhend, URL [taltech-coding.gitlab.io](http://taltech-coding.gitlab.io/) [12.09.2023]

# Kokkuvõtte

Oli tehtud rakendus mis nimetus on „Veebipood”, selles rakenduses on võimalus lisada, kustutada ja uuendada andmeid. Mina usun, et see rakenduse on võimalus veel areneda ja edasi kuskil kasutada nagu veebipood.