Tallinna Tööstushariduskeskus

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

**Õpilased:** Daniil Kuzjomin  
 **Rühm:** TARpv21  
 **Juhendaja:** IT kutseõpetaja Marina Oleinik

Tallinn 2023

Sisukord

1. Sissejuhatus

**Projekti eesmärk:** projekti eesmärk on luua veebirakendus ORM kasutades, veebirakendus “Veebipood” milles kasutaja saab salvestada andmeid andmebaasis jne.

**Projekti asjakohasus, tähtsus, vajalikkus:** aitab veebirakenduse kasutajal andmebaasi andmeid lisada ja salvestada.

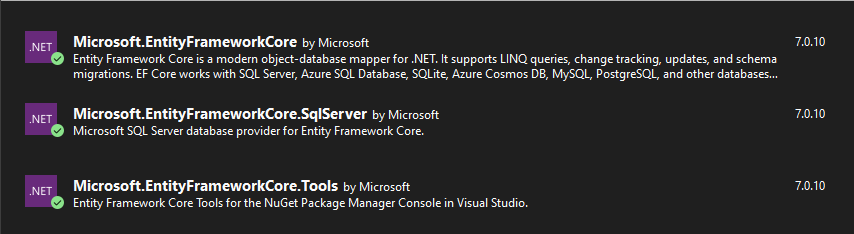
**Sihtpublik:** kasutajad, kes kasutavad andmete lisamiseks veebirakendust

**Nõutav tehnika tööriistad projekti loomiseks:**

* Arvuti, kuhu on installitud Visual Studio arenduskeskkond.
* Seadistatud Microsoft EntityFrameWorkCore moduleid projektiga.
* Postman rakendus, et töötada andmebaasiga.

1. Seadistamine

1 punkt, mida pidin tegema, oli lisada uued moodulid Visual Studio projektile, et seda teha, ma avasin oma uus projekt ja järgin seda tee “Tools - NuGet Package Manager – Manage NuGet Packages for Solution... ”. Pärast avamist, ma installin järgmised mudelid, millised on järgmist screenshotist:



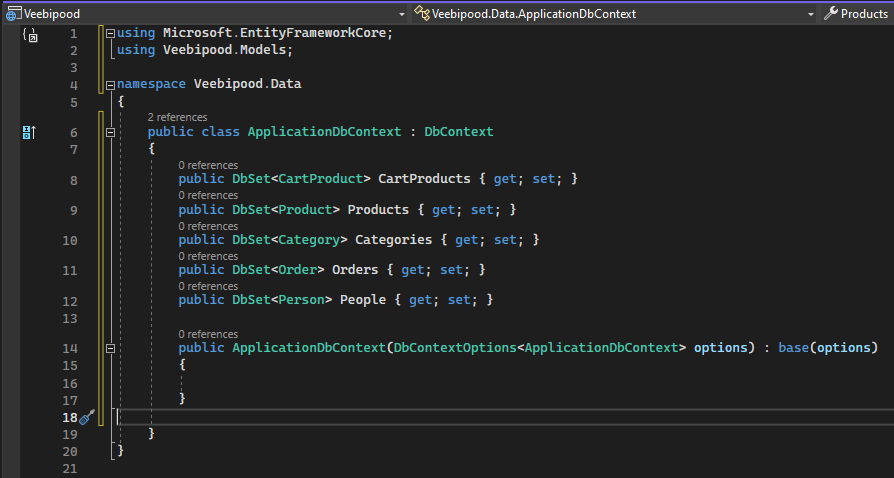
Pärast installimist ma järgin teise punktisse.

**ApplicationDbContext loomine**

Looge projektis kaust Data - looge uus klass ja nimetage seda ApplicationDbContext.cs on andmebaasi konfiguratsioonifail, milles määratleme kõik tabelid, mis meil on:



Määrame meie andmebaasi kõik tabelid selles classis nagu screenshotis:



**Andmebaasi aadress lisamine**

Täpsustame oma andmebaasi aadressi, selleks kasutame seda vihjet "vaade" -> "scl server object explorer" -> nooleste aastat vazhutada skl server -> localdb jõi paremaid klippe + nimeta ümber -> kopeeri"

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

"ConnectionStrings": {

"DefaultConnection": "server=;database=Pood;"

}

Lisame server väljal varem kopeeritud andmed

Database väljal kirjutame meie andmebaasi nimi, ma valisin “Pood” sest teen veebipood.

Lõppes meie appsettings.json fail on järgnev:

{

"ConnectionStrings": {

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

},

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft.AspNetCore": "Warning"

}

},

"AllowedHosts": "\*"

}

**Seadistus Program.cs failis**

Järgmine punkt on lisada järgmised read Program.cs failisse

Leiame Program.cs fail ja avame. Avatud failisse on vaja lisada järgmised read:

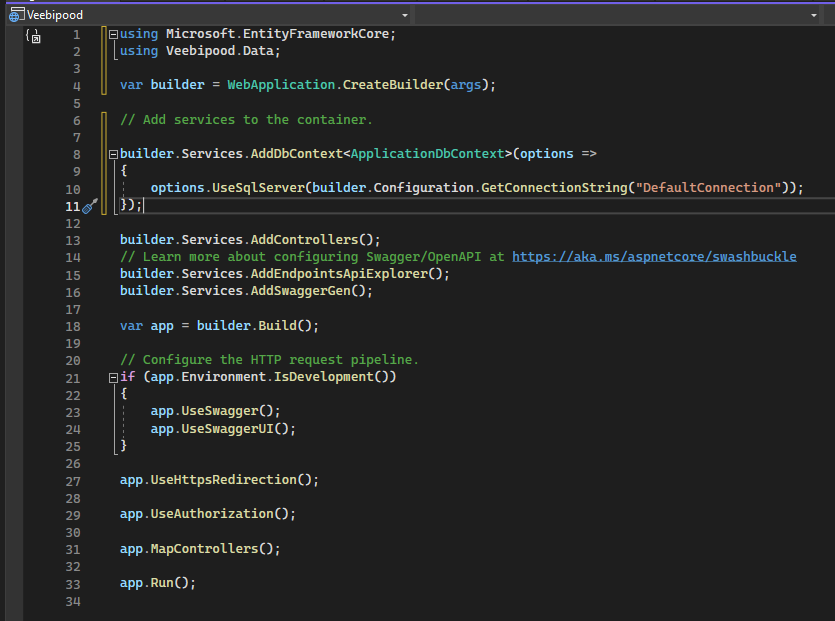
builder.Services.AddDbContext<ApplicationDbContext>(options =>

{

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

});

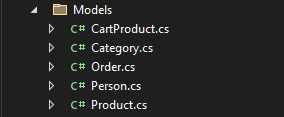
Neid on vaja meile, et seadistada andmebaas. Pärast reade lisast Program.cs fail on järgnev:



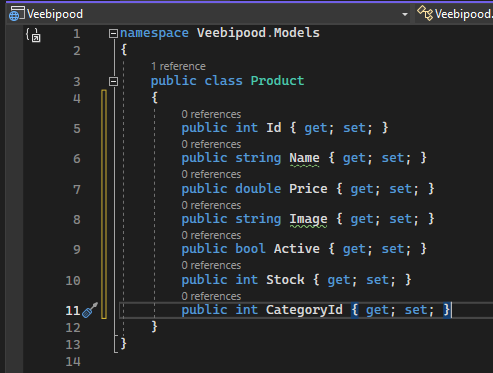
**Mudeli koostamine**

Pärast Program.cs redigeerimist, lisame kõik vajalikud mudelid

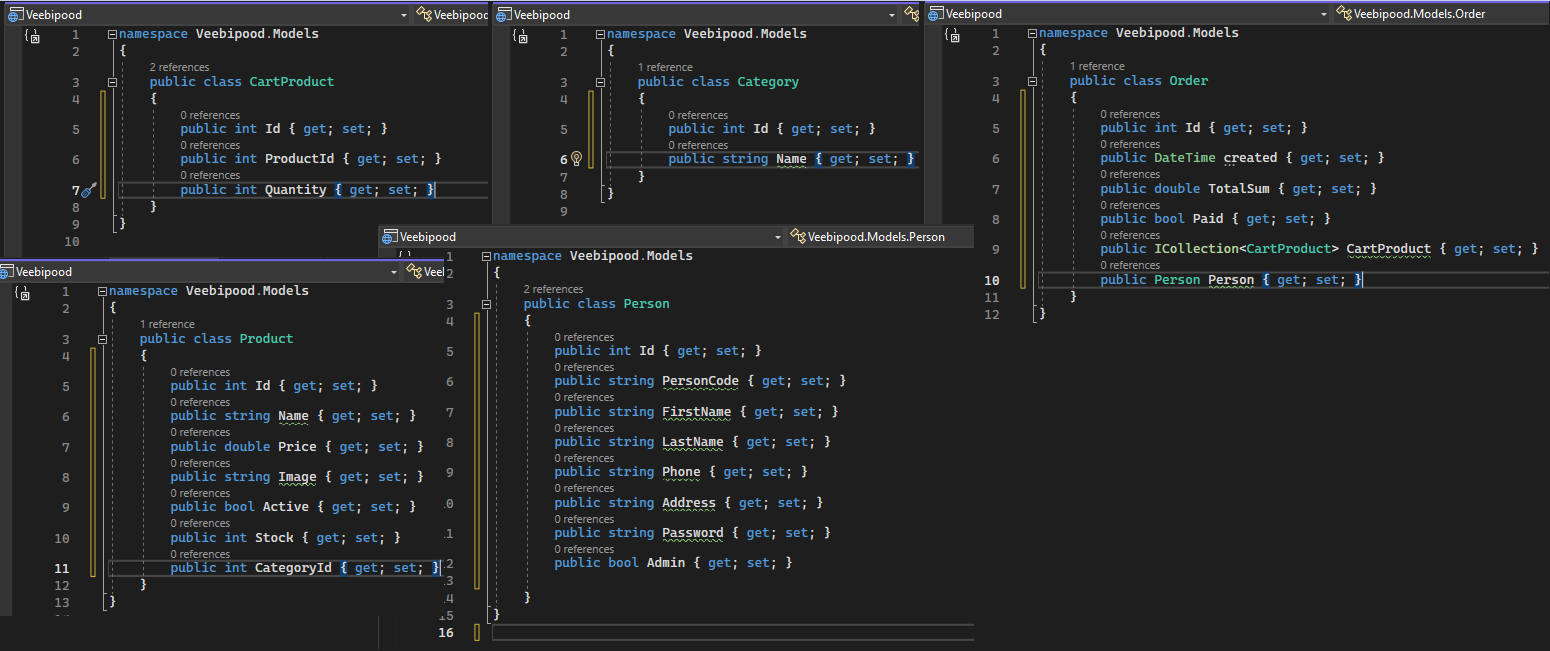
Sellest, me teeme uus kaust nimega “Models” ja lisame sellese kaustikule kõik mudelid. Pärast classide lisamist minu kaust on järgnev:

  
Pärast, avame mudelid ja teeme andmebaasi mudelite vorm, muutujate ja nende tüüpide kujul

Näide:

  
Üks mudelitest kus ma tegin vormi.

Lõppes kõik tekkinud mudelid on järgnevad:



**Lisame mudelid andmebaasile**

Nüüd meil on vaja lisada meie mudelid andmebaasile. See on väja teha iga kord kui me loome uus mudel ja tegime selle mudeli vorm.

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

Pärast, me kirjutame päringut millised näitavad millised mudelid me tegime.

Mul oli vaja sellised päringut:

add-migration addedCartProduct

add-migration addedCategory

add-migration addedOrder

add-migration addedPerson

add-migration addedProduct

Pärast seda, kui me kirjutame kõik päringut ja need töötasid, meil on vaja uuendada andmebaas, meil on vaja sellest järgmine päring:

update-database

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

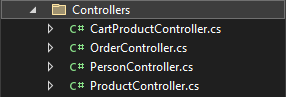
"Add-Migration UpdateColumnName"

Ja pärast seda paringu, on vaja kirjuta andmebaasi uuendamise päring ka.

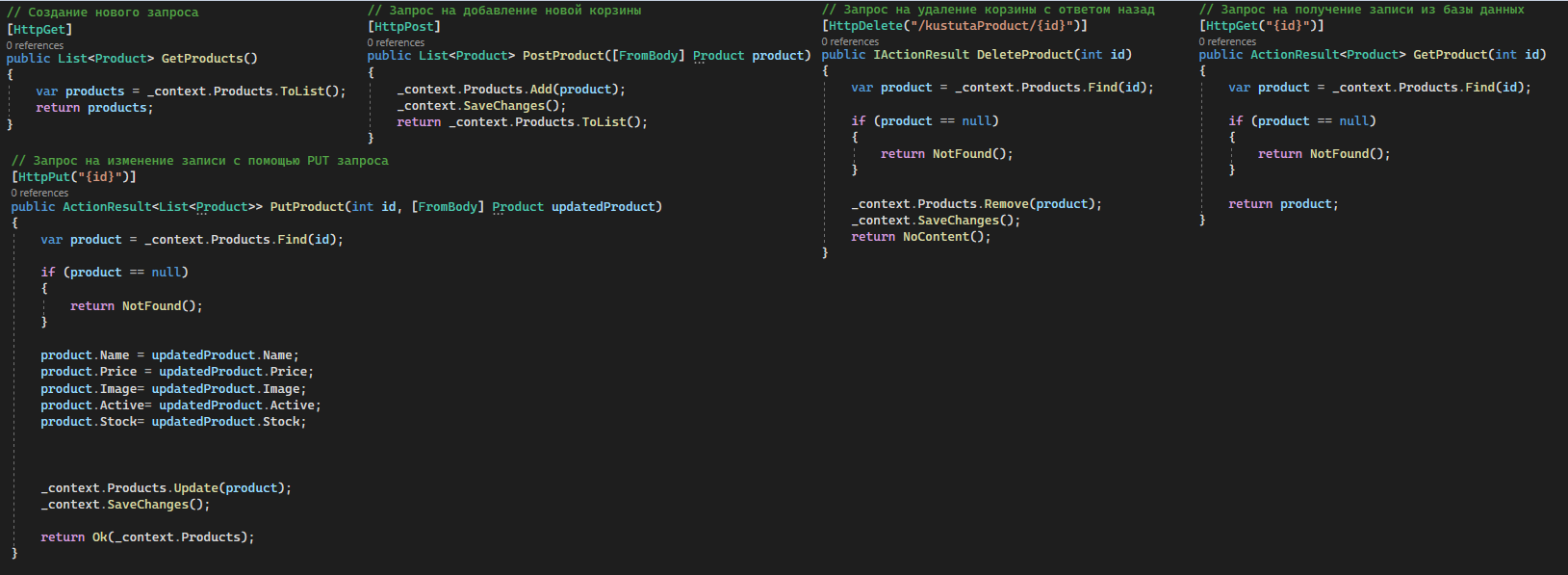
**Kontrollerid**

Pärast tegemist andmebaasis kõik muudatused päringuga, peame looma kontrollerid. Siis meil on vaja “Controllers” kaust ja teha selles kaustis kontrollerid klõpsame kaustile vasaku hiire nupp - Add – Controllers.

Lisame kõik meil vaja kontrollerid, lõppes mul see valja näeb selline:



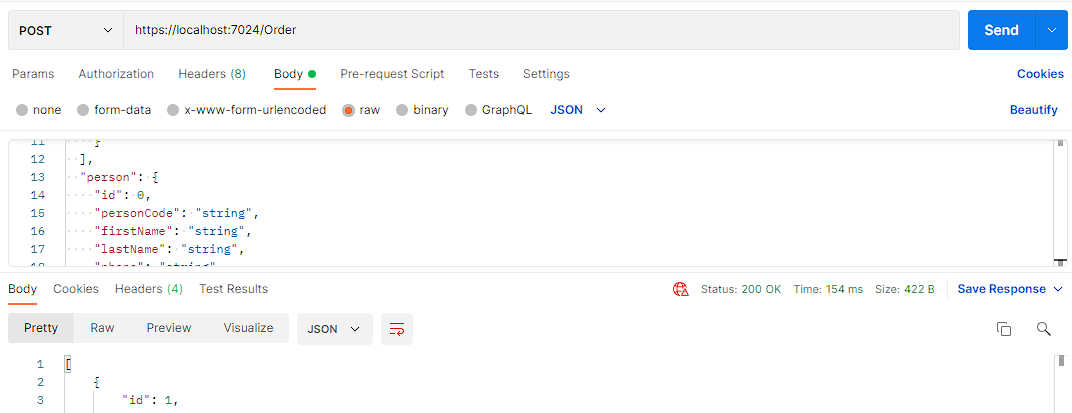
Nüüd konfigureerime need kontrollerid. Sellest ma tegin järgmised päringud:

  
Igal kirjutatud päringutel ma tegin komentaar mis see päring teeb.

**Andmebaasi töö kontroll.**

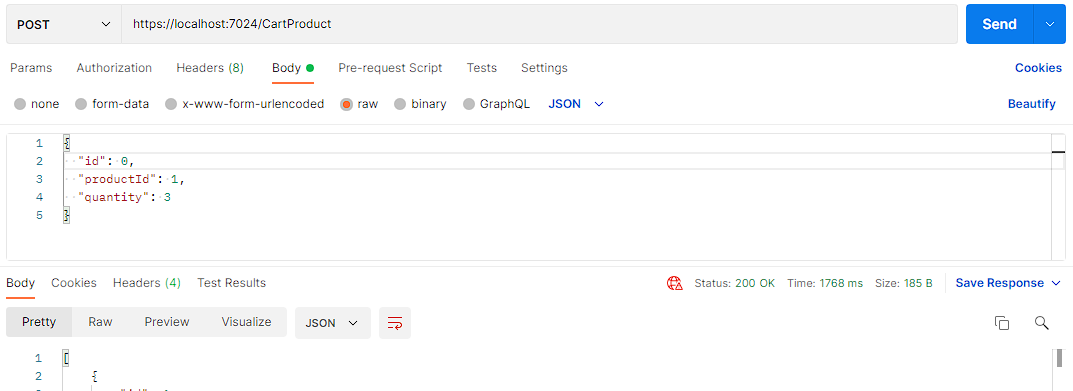
Andmebaasi kontrollimiks meil on vaja “Postman” rakendus. Avame seda, ja nüüd meil on vaja meie localhosti link. Meil on vaja käivita meie rakendus, ja brauseris kopeerida selline link, nagu screenshotis:

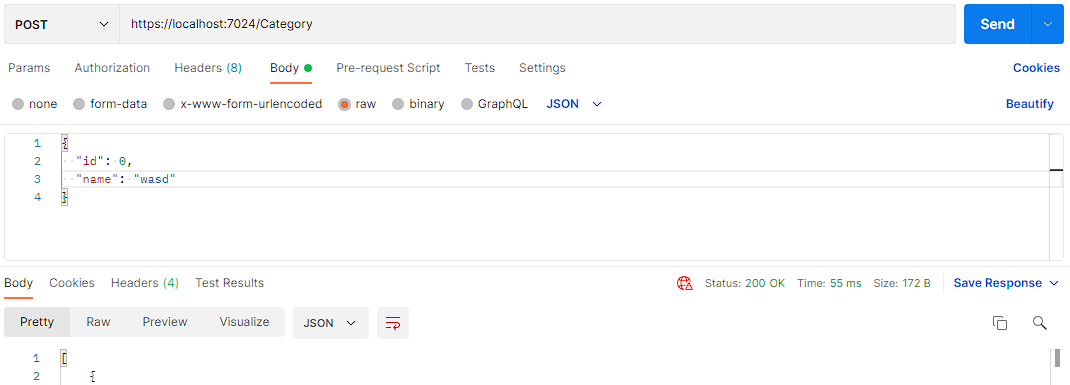


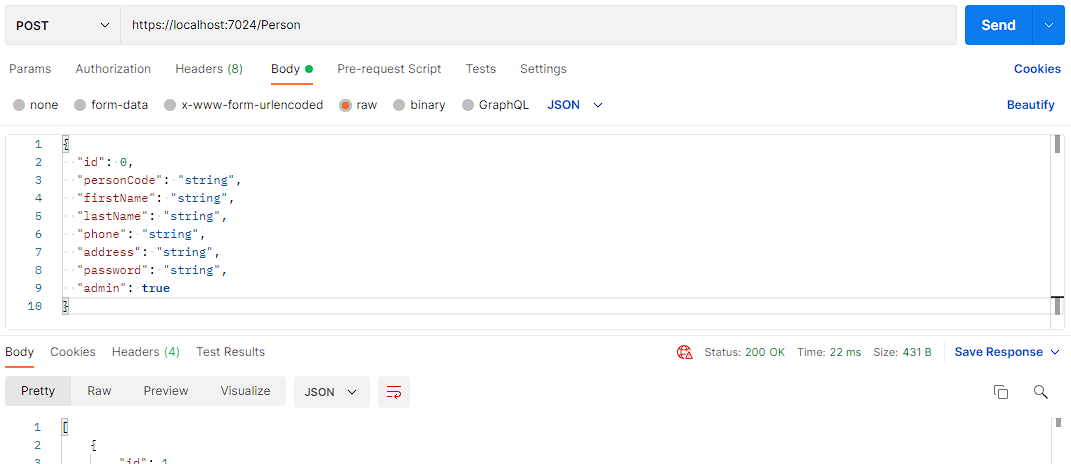
Nüüd meil on link, seda on vaja panna Postmani rakendusele, ja siis saame proovida kontrollida meie andmebaas, mul on järgmised päringud oli tehtud:  


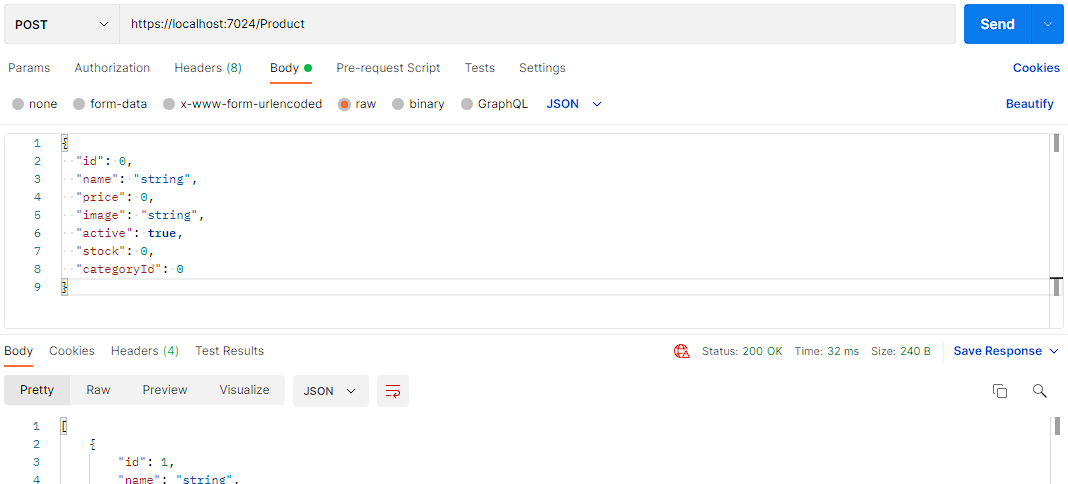
Siin ma proovin teha “Order” päring, ja kõik töötab

Nii saama ma tegin kõigiga päringuga:









Kõik päringud töötavad, see tähendab et tegin kõik õigesti.

1. Kasutatud kirjeldus / allikad
2. TalTech juhend, URL [taltech-coding.gitlab.io](http://taltech-coding.gitlab.io) [05.09.2023]
3. Kokkuvõtte

Oli tehtud veebirakendus, millega saad lisada, redigeerida, kustuta andmeid andmebaasist. Tulevikus arvan, et kasutajad saab kasutaja seda poele.