# Uputstvo za kreiranje VS 2022 .NET5 projekta

Uverimo se da imamo instalirane:

* Visual Studio 2022
  + https://visualstudio.microsoft.com/vs/

* Visual Studio Web development paket
  + Idemo na: Install more toolas and features (u Create a new project prozoru)
  + Stikliramo ASP.NET and web development
* .NET 5.0
  + Idemo na: Install more toolas and features (u Create a new project prozoru)
  + Izaberemo karticu Individual components
  + Stikliramo .NET 5.0 Runtime a zatim klik na dugme Install while downloading

U Visual Studio-u 2022 izaberemo Create a new project > ASP.NET Core Web API.

Mozemo koristiti I ASP.NET Core Web Empty ali nam ASP.NET Core Web API projekat obezbedjuje Swagger UI, koji je vrlo koristan za testiranje I vizuelizaciju osnovnih CRUD operacija koje ce kasnije biti potrebne za komunikaciju sa bazom.

Izaberemo naziv I lokaciju projekta a zatim .NET 5 iz padajuceg menija.

# Modalne klase

U okviru Solution Explorer-a unutar trenutnog projekta kreiramo novi folder u koji cemo smestati modalne klase.

Ove klase ce kasnije kada dodjemo do baza omoguciti I migraciju baze, a trenutno cemo ih koristiti samo za komunikaciju sa Frontend-om.

Kreiramo najobicniju klasu koja ce predstavljati objekat koji se prenosi (mora biti identicna objektu koji primamo/saljomo na frontend) I kreiramo konstruktor sa svim parametrima.

Primer:

public class Uplata

{

public Uplata(int id, int iznos, string valuta, string datum)

{

Id = id;

Iznos = iznos;

Valuta = valuta;

Datum = datum;

}

public int Id { get; set; }

public int Iznos { get; set; }

public string Valuta { get; set; }

public string Datum { get; set; }

}

# Kontroleri

Kontroleri su neophodni za bilo kakvu direktnu komunikaciju sa Frontend-om, odnosno za primanje GET I POST zahteva od naseg Frontend-a.

U postojecem folderu Controllers kreiramo novi kontroler.

Na novom prozoru na levoj strani izaberemo API a zatim na sredini prozora API-Controller Empty.

Naziv samog kontrolera (klase) predstavlja url za POST,GET,DELETE,PUT zahteve.

Primer: igraciController.cs ==> api/igraci

Zahtevi se salju/primaju putem metoda koje su oznacene u zavisnosti koju funkciju obavljaju:

**[HttpGet]** - uzima sve clanove/listu klase

**[HttpGet(“{id}”)]** - uzima clana klase sa odredjenim id

**[HttpPost]** - preuzima podatke “u .NET”

**[HttpPut("{id}")]** - postavlja vrednost odredjenog clana liste neke klase (update za odredjeni element sa odredjenim id-em)

**[HttpDelete("{id}")] -** brisanje elementa liste klase sa odredjenim id-em

# CORS – Cross Origin Resource Sharing

CORS nam omogucava da “pruzimo dozvolu” nasem frontendu da komunicira sa url-ima koje smo prethodno kreirali preko kontrolera.

CORS se u .NET 5 podesava na sledeci nacin:

* Otvorimo **Startup.cs**, a zatim pronadjemo public void funkciju **ConfigureServices**
* Na **kraju funkcije** (unutar funkcije) dodamo liniju **AddCors()** koja omogucava CORS
* U void funkciji **Configure** na **pocetku funkcije** dodamo sledece linije:

App.UseCors(options =>

Options.WithOrigins(“http://localhost:4200”)

.AllowAnyMethod()

.AllowAnyHeader());

U Options.WithOrigins() unosimo **url naseg frontend-a** da bi backand znao odakle moze da prima zahteve a odakle ne moze (radi bezbednosti).