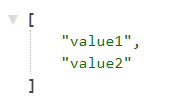
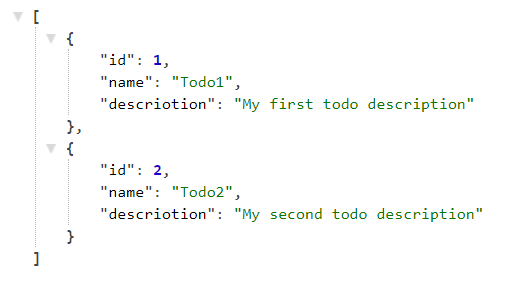
# **NMA WebApi kūrimo workshop‘as**

1. Susikurkite naują aplanką (folder), kuriame kursime projektą.
2. Atsidarykite CMD (Command Prompt) arba powershell.
3. Nueikite į naujai sukurtą aplankalą naudojant cd komandą   
   *cd C:\Users\adavdaj\Desktop\nma*
4. Patikrinkite ar dotnet komanda pasiekiama jūsų kompiuteryje  
   *dotnet --info*  
   Komanda turėtų pavaizduoti įrašytą dotnet versiją, pavyzdžiui:  
   *2.2.202*
5. Sukuriame naują webapi projektą:  
   *dotnet new webapi*  
   Komanda automatiškai sukuria webapi projektą su tokiu pačiu vardu, kuriuo užvadinote aplankalą. Tai pakeisti galima pasinaudojus papildomais parametrais.  
   Norint gauti daugiau informacijos apie komandą new ir jos parametrus, įveskite:  
   *dotnet new --help*
6. Atsidarome VSCode arba kokį nors kitą Code Editorių.  
   *code .*
7. Įsitikinkime, kad kodas yra be loginių klaidų ir kad mūsų kompiuteris sugeba kodą sukompiliuoti:  
   *dotnet build*Jeigu komanda sėkminga – turėtumėte matyti užrašą: Build succeeded.Galima naudoti integruotą VSCode terminalą panaudojus Ctrl + ` arba toliau naudoti cmd/powershell.
8. Norint paleisti WebApi – paspaudžiame F5.
9. Kadangi VSCode nežino kokiai aplinkai skirta aplikacija, pirmo paleidimo metu paprašoma ją pasirinkti. Pasirenkame .NET Core.
10. Atsidaro sugeneruotas launch.json failiukas. Galime peržvelgti reikšmes, tačiau nieko keisti neturėtų reikėti.
11. Spaudžiame F5 dar kartą.
12. Šį kartą atsidaro interneto naršyklė ir matome langą su žinomiausiu HTTP status kodu – 404.
13. Paredaguokite URL pridedami /api/values į pabaigą.
14. Turėtumėte matyti duomenų rezultatą  
    
15. Norint pamatyti kaip šie duomenys atsirado, atsidarykite Controllers aplankale esantį ValuesController.
16. Sukuriame modelio klasę – TodoItem.cs
17. Sukuriame tris properties – int ID, string Name, string Description.
18. Sukuriame naują Controller klasę pavadinimu TodoController.cs
19. Susikuriame naują statinį klasės kintamąjį, kuriame saugosim sąrašą TodoItem.
20. Inicializuojame kelis objektus.
21. Pakeičiame Get metodą taip, kad grąžintu ne string, o TodoItem.
22. Paleidžiame aplikaciją su F5 ir atsidarome /api/todo URL.  
    Turėtumėme matyti kažką panašaus į  
    
23. Pakeičiame metodą POST taip, kad pridėtų naują objektą į sąrašą prie TodoItems.
24. Pakeičiame likusius HTTP metodus, kad atliktų atitinkamus veiksmus.
    1. Get (su id) grąžintų rastą elementą.
    2. Put sukurtų naują elementą jeigu tokio nėra ir pakeistų jeigu jau yra sukurtas.
    3. Delete panaikintų elementą.
25. Pridedam Swagger json dokumento generavimą bei SwaggerUI dokumento atvaizdavimui pagal dokumentaciją: <https://github.com/domaindrivendev/Swashbuckle.AspNetCore>

Turim veikiantį WebApi, kuris realizuoja CRUD operacijas ir panaudoja Swagger WebApi dokumentavimą.