Using EF Core in an ASP.NET Core API

Julie Lerman
MOST TRUSTED AUTHORITY ON ENTITY FRAMEWORK
@julielerman thedatafarm.com

Module Overview

- Create an ASP.NET Core API project
- Add in a controller for the Samurais
- Wire up ASP.NET Core to SamuraiContext
- Learn a few tricks for debugging and logging
- Interact with controller methods to read and write data
- Work with a great tool for sending HTTP commands and data to the controller for basic testi ng

Adding the ASP.NET Core Project

Adding a Samurai Controller to the Project

Wiring Up ASP.NET Core with EF Core

❖ Web API Project

1. Add references to projects with entities and DbContext
2. Build
3. Add controller (EF and API actions)
   (This will add EF Core and other packages to csproj)

Wiring up the ASP.NET Core App with the Our DbContext

Wiring up ASP.NET Core with EF Core

- **Web API Project**
  1. Add references to projects with entities and DbContext
  2. Build
  3. Add controller (EF & API actions) *(Adds EF Core & other packages to csproj)*
- **Startup**

1. 4. Add services.DbContext with UseSqlServer to startup Configure()

- **appsettings.json**
  5. Add connection string config
  6. Add EF Core logging config
- **DbContext**
  7. Add constructor that takes in DbContextOptions
  8. Set ChangeTracker.QueryBehaviorOptions to NoQuery in constructor
  9. Remove optionsBuilder from OnConfiguring
  10. Remove ConsoleLoggerFactory
  11. Clean up using statements

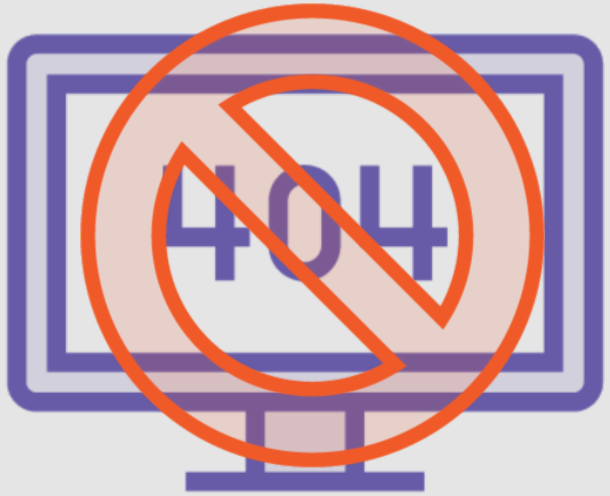
Wiring up ASP.NET Core with EF Core

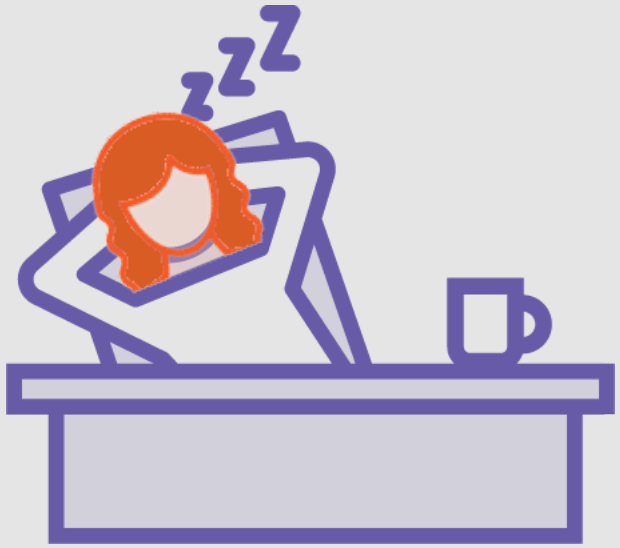
- **Web API Project**
  1. Add references to projects with entities and DbContext
  2. Build
  3. Add controller (EF & API actions) *(Adds EF Core & other packages to csproj)*
- **Startup**
  4. Add services.DbContext with UseSqlServer to startup Configure()
- **appsettings.json**
  5. Add connection string config
  6. Add EF Core logging config
- **DbContext**
  7. Add constructor that takes in DbContextOptions
  8. Set ChangeTracker.QueryBehaviorOptions to NoQuery in constructor
  9. Remove optionsBuilder from OnConfiguring
  10. Remove ConsoleLoggerFactory
  11. Clean up using statements

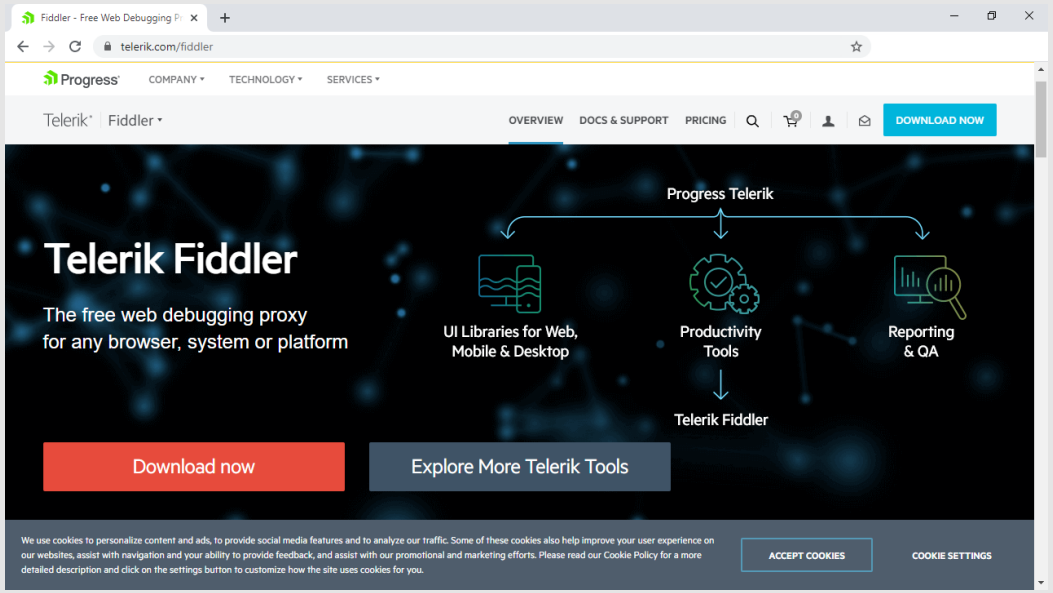
Running the Controller to See the Output

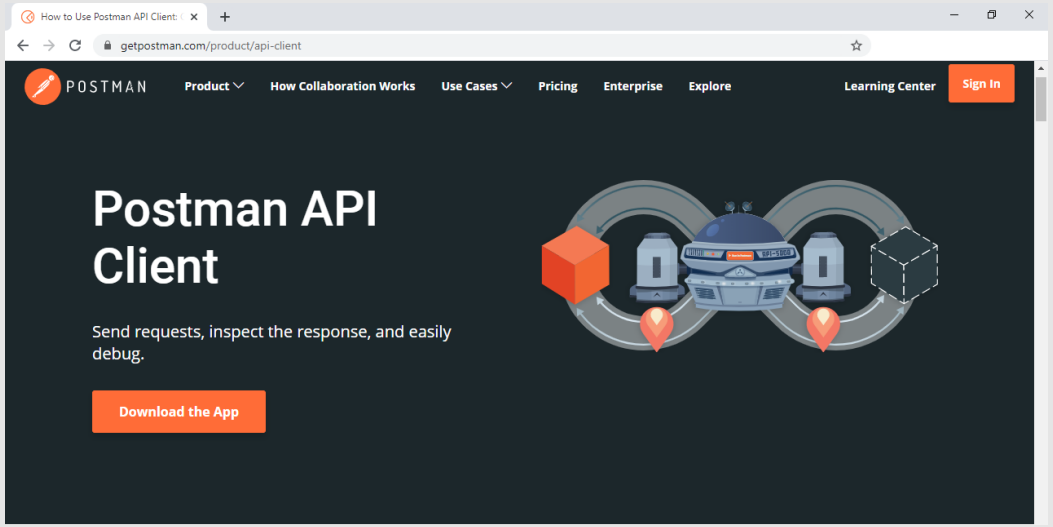
Exploring and Debugging Insert, Update & Delete Controller Methods

How Will We Run the PUT POST & DELETE?









Review

- Created a new ASP.NET Core API project
- Referenced Data & Domain projects, *built* the solution & created a controller
- Modified startup, appsettings & DbContext to integrate API & EF Core
- Learned about the logging integration
- Learned how to use REST Client to send JSON to HTTP requests
- Created a new method that uses our stored procedure and works with related data

Up Next!

Testing with the InMemory provider instead of a database

Resources

Entity Framework Core on GitHub github.com/aspnet/entityframework

EF Core Documentation docs.microsoft.com/ef

Visual Studio Code homepage: code.visualstudio.com

JSON View extension for Google Chrome: bit.ly/jsonviewext

REST Client extension for Visual Studio Code (also find it in VS Code IDE)
marketplace.visualstudio.com/items?itemName=humao.rest-client

SOLID Principles for C# Developers Course
app.pluralsight.com/library/courses/csharp-solid-principles/table-of-contents

Using EF Core in an ASP.NET Core API

Julie Lerman
MOST TRUSTED AUTHORITY ON ENTITY FRAMEWORK
@julielerman thedatafarm.com