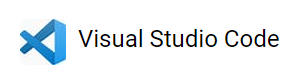
Generate API Docs : Swagger

GENERATE API DOCS

SWAGGER

Muhammad Arif Rahman - FSDO003ONL004 – Dokumentasi Sesi 11



1. Aplikasi persiapan
   1. Visual Code
   2. Browser
   3. Xampp
   4. Docker
   5. VS Code
   6. SQLyog
   7. Postman

buat baru

dotnet new webapi -n "Payments" -lang "C#" -au none

library

dotnet add package NuGet.CommandLine.XPlat --version 5.11.0

dotnet add package Microsoft.EntityFrameworkCore.Tools --version 5.0.11

dotnet add package Microsoft.EntityFrameworkCore.Sqlite --version 5.0.11

dotnet add package Microsoft.EntityFrameworkCore.Design --version 5.0.11

library jwt

dotnet add package Microsoft.AspNetCore.Authentication.JwtBearer --version 5.0.11

dotnet add package Microsoft.AspNetCore.Identity.EntityFrameworkCore --version 5.0.11

dotnet add package Microsoft.AspNetCore.Identity.UI --version 5.0.11

menambah database

dotnet ef migrations add "Initial Migrations"

dotnet ef database update

<https://www.browserling.com/tools/random-string> = set 32

secret jwt config in appsetting.json

dotnet ef migrations add “Addig authentication to our Api” generate=bin

dotnet ef database update

jika update ApiDbContext.cs

dotnet ef migrations add "Added refresh tokens table"

dotnet ef database update

dotnet add package MySql.Data --version 8.0.28

dotnet add package MySql.EntityFrameworkCore --version 5.0.8

MySql.EntityFrameworkCore

<https://docs.google.com/forms/u/0/d/e/1FAIpQLSca9QDX4Ufwr7lqvRxVqIg4nQYUQpjA9_fzoRY6nok7-FnM8Q/formResponse?pli=1>

using TodoAppWithJWT.Configuration;using Microsoft.AspNetCore.Authentication.JwtBearer;using Microsoft.AspNetCore.Identity.EntityFrameworkCore;using Microsoft.AspNetCore.Identity.UI;using System.Text;using Microsoft.IdentityModel.Tokens;using Microsoft.AspNetCore.Identity;

c.AddSecurityDefinition("Bear

c.AddSecurityRequirement(new OpenApiSecurityRequirement() { { new

var key = Encoding.ASCII.GetBytes(Configuration["JwtConfig:Secret"]); var TokenValidationParams = new TokenValidationParameters { ValidateIssuerSigningKey = true, IssuerSigningKey = new SymmetricSecurityKey(key), ValidateIssuer = false, ValidateAudience = false, ValidateLifetime = true, RequireExpirationTime = false, ClockSkew = TimeSpan.Zero };

services.AddSingleton(TokenValidationParams); services.AddAuthentication(options => { options.DefaultAuthenticateScheme = JwtBearerDefaults.AuthenticationScheme; options.DefaultScheme = JwtBearerDefaults.AuthenticationScheme; options.DefaultChallengeScheme = JwtBearerDefaults.AuthenticationScheme; }) .AddJwtBearer(jwt => { jwt.SaveToken = true; jwt.TokenValidationParameters = TokenValidationParams; }); services.AddDefaultIdentity<IdentityUser>(options => options.SignIn.RequireConfirmedAccount = true) .AddEntityFrameworkStores<ApiDbContext>();

