

Documentatie cloud API

In dit document ga ik de 2 API uitbereidingen verklaren.

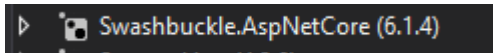
API uitbereidingen

OpenAPI

Open API is een automatisch gegenereerde documentatie van je API. Deze wordt dan als de index pagina van de API gezet, zodat de mensen die de API willen gebruiken gemakkelijk kunnen zien welke requests ze allemaal kunnen doen met deze API. Men kan ook requests testen via deze pagina en er staat ook hoe de schema's zijn opgebouwd.

Installatie:

NuGet package:



Implementatie:


```
0 references | Niels verschooten, 19 hours ago | 1 author, 6 changes
public void ConfigureServices(IServiceCollection services)
{
    services.AddAuthentication(JwtBearerDefaults.AuthenticationScheme)
        .AddJwtBearer(options =>
        {
            options.Authority = "https://securetoken.google.com/cloud-api-2021";
            options.TokenValidationParameters = new TokenValidationParameters
            {
                ValidateIssuer = true,
                ValidIssuer = "https://securetoken.google.com/cloud-api-2021",
                ValidateAudience = true,
                ValidAudience = "cloud-api-2021",
                ValidateLifetime = true
            };
        });
    services.AddDbContext<ESOTrialContext>({
        options => options.UseSqlServer(
            Configuration.GetConnectionString("DefaultConnection")
        )
    });
    services.AddMvc();
    services.AddControllers().AddNewtonsoftJson(options => options.SerializerSettings.ReferenceLoopHandling = Newtonsoft.Json.ReferenceLoopHandling.Ignore);
    services.AddSwaggerGen(c =>
    {
        c.SwaggerDoc("v1", new OpenApiInfo { Title = "ESOTrialAPI", Version = "v1" });
    });
}


0 references | Niels verschooten, 19 hours ago | 1 author, 3 changes
public void Configure(IApplicationBuilder app, IWebHostEnvironment env, ESOTrialContext eSOTrialContext)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    DBInitializer.Initialize(eSOTrialContext);
    app.UseRouting();
    app.UseSwagger();
    app.UseSwaggerUI(c =>
    {
        c.SwaggerEndpoint("/swagger/v1/swagger.json", "ESOTrialAPI");
        c.RoutePrefix = string.Empty;
    });
    app.UseCors(builder =>
        builder.AllowAnyOrigin()
            .AllowAnyMethod()
            .AllowAnyHeader());
    app.UseAuthentication();
    app.UseAuthorization();

    app.UseEndpoints(endpoints =>
    {
        endpoints.MapControllers();
    });
}
```

URL: <https://docs.microsoft.com/en-us/aspnet/core/tutorials/web-api-help-pages-using-swagger?view=aspnetcore-5.0>

Resultaat:

 **Swagger**
Supported by SMARTBEAR

Select a definition **ESOTrialAPI** 

ESOTrialAPI v1 OAS3

/swagger/v1/swagger.json

Enchantment

GET /api/v1/enchantments

item

GET /api/v1/items

GET /api/v1/items/{id}

DELETE /api/v1/items/{id}

GET /api/v1/items/weapons


POST /api/v1/items/weapons

PUT /api/v1/items/weapons

GET /api/v1/items/armors

POST /api/v1/items/armors

PUT /api/v1/items/armors



EF Migration

EF migration is een manier om je database in sync te houden op al je applicaties, Dit gebeurt door files die gegenereerd worden die de database en alles wat er inzit describen. Deze files kunnen dan gebruikt worden om de andere databases up te daten. Ik denk dat je deze files ook kan gebruiken als een soort back up voor je database, maar daar ben ik niet zo zeker van.

Implementatie:

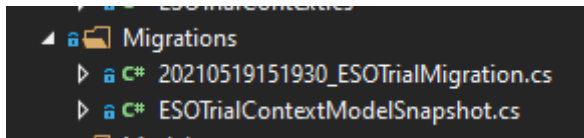
Je hebt de .NET Core CLI Tools nodig en daarna is het allemaal commando's in de commandline

Create new migration:

```
dotnet ef migrations add ESOTrialMigration
```

Update database:

```
dotnet ef database update
```



URL: <https://docs.microsoft.com/en-us/ef/core/managing-schemas/migrations/?tabs=dotnet-core-cli>