

Eigenlijk wil je `Install-Package MySql.EntityFrameworkCore` maar deze is nog niet compatible met Core 6.0

Dus

Gebruik je

`Install-Package Pomelo.EntityFrameworkCore.MySql`

Deze is al wel compatible met 6.0

```
Install-Package Microsoft.EntityFrameworkCore.Tools
```

Appsettings.json:

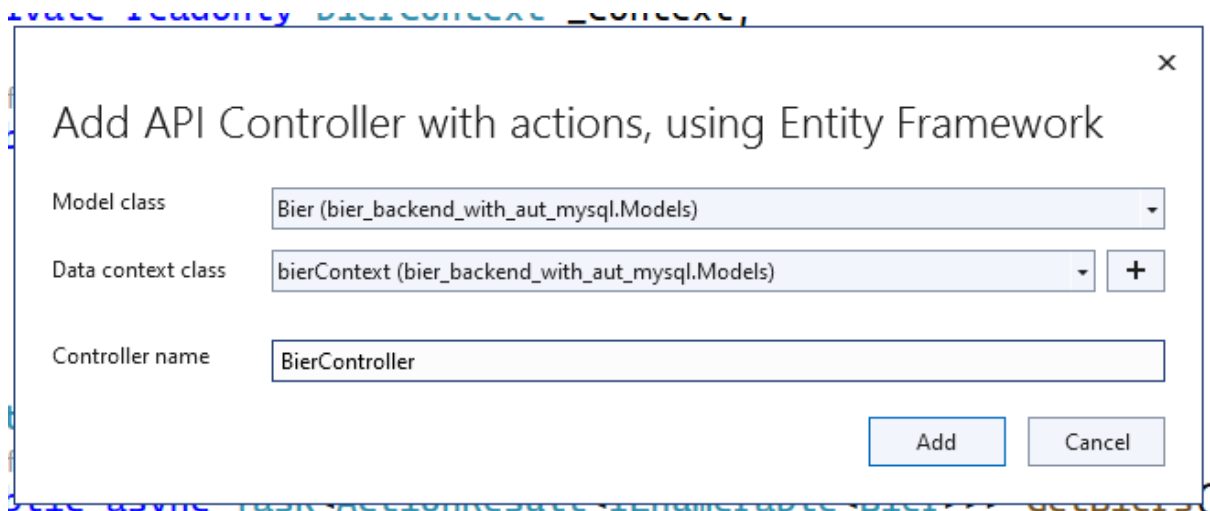
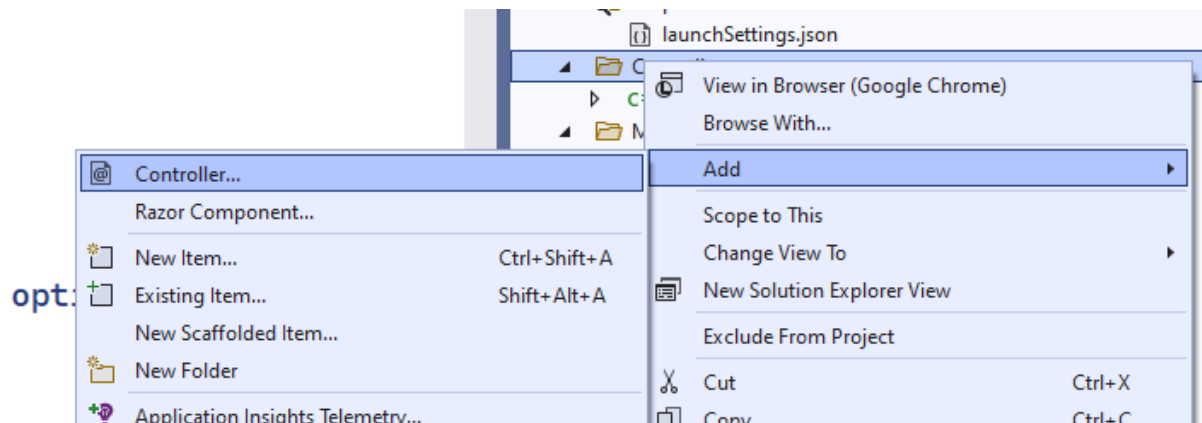
```
Server=localhost;Database=bier;Uid=root;Pwd=myPassword;
```

```
Scaffold-DbContext name=ConnectionStrings:MySQLDb  
Pomelo.EntityFrameworkCore.MySql -OutputDir Models -f
```

Aan onze Program.cs voegen we weer de connectie met DB code toe:

```
ConfigurationManager config = builder.Configuration;  
  
var connectionString = config.GetConnectionString("MySQLDb");  
var serverVersion = ServerVersion.AutoDetect(connectionString);  
  
builder.Services.AddDbContext<bierContext>(options =>options  
    .UseMySQL(connectionString,serverVersion )  
    // The following three options help with debugging, but should  
    // be changed or removed for production.  
    .LogTo(Console.WriteLine, LogLevel.Information)  
    .EnableSensitiveDataLogging()  
    .EnableDetailedErrors()  
    );
```

NU kunnen we een Controller aanmaken voor het aangemaakte Bier Model.:



En de laatste stap is nu om JWT Bearer token Authenticatie toe te voegen.

Belangrijk verschil met de vorige applicatie is dat we nu al de Bier database hebben en in een context hebben zitten , waarbij er ook al data in de database zit.

We kunnen dus niet migrations doen op deze context dan raken we de data kwijt, of krijg je foutmeldingen dat tabellen al bestaan.

De oplossing is om 2 DbContexts te hebben, 1 automatisch gegenereerd (bierContext) en de zelf gemaakte ApplicationDbContext voor Identity en inloggen. Beiden maken ze gebruik van dezelfde db connection string:

```
builder.Services.AddDbContext<bierContext>(options =>options
    .UseMySQL(connectionString,serverVersion )
    // The following three options help with debugging, but should
    // be changed or removed for production.
    .LogTo(Console.WriteLine, LogLevel.Information)
    .EnableSensitiveDataLogging()
    .EnableDetailedErrors()
);
//2e context voor de gebruikers gegevens. Anders gaan migrations niet goed (
//overschrijft ie de biertabel)
builder.Services.AddDbContext<ApplicationDbContext>(options => options
    .UseMySQL(connectionString, serverVersion)
    // The following three options help with debugging, but should
    // be changed or removed for production.
    .LogTo(Console.WriteLine, LogLevel.Information)
    .EnableSensitiveDataLogging()
    .EnableDetailedErrors()
);
```

Stappen zijn weer nu ongeveer hetzelfde als het vorige project, we hebben db tabellen nodig om te users rollen en wachtwoorden in op te slaan. En een controller om te kunnen in loggen en te registreren zodat we ook echt het bearer token krijgen.

Even stap voor stap:

- Installeer Package : **Microsoft.AspNetCore.Authentication.JwtBearer**

Toevoegen aan de Program.cs de configuratie:

```
// For Identity
builder.Services.AddIdentity<IdentityUser, IdentityRole>()
    .AddEntityFrameworkStores<ApplicationDbContext>()
    .AddDefaultTokenProviders();

// Adding Authentication
builder.Services.AddAuthentication(options =>
{
    options.DefaultAuthenticateScheme =
JwtBearerDefaults.AuthenticationScheme;
    options.DefaultChallengeScheme =
JwtBearerDefaults.AuthenticationScheme;
    options.DefaultScheme = JwtBearerDefaults.AuthenticationScheme;
}))

// Adding Jwt Bearer
.AddJwtBearer(options =>
{
    options.SaveToken = true;
    options.RequireHttpsMetadata = false;
    options.TokenValidationParameters = new TokenValidationParameters()
    {
        ValidateIssuer = true,
        ValidateAudience = true,
        ValidAudience = configuration["JWT:ValidAudience"],
        ValidIssuer = configuration["JWT:ValidIssuer"],
        IssuerSigningKey = new
SymmetricSecurityKey(Encoding.UTF8.GetBytes(configuration["JWT:Secret"]))
    };
});
```

Add-Migration Initial -Context ApplicationDbContext

Update-Database -Context ApplicationDbContext

En op deze manier wordt alleen de migration gedraait van de ApplicationDbContext met alleen de tabellen voor identity: aspnetroles, aspnetusers, etc.

localhost - localhost:3306	
Databases	
bier	
Tables	
> __efmigrationshistory	16K
> aspnetroleclaims	48K
> aspnetroles	32K
> aspnetuserclaims	48K
> aspnetuserlogins	48K
> aspnetuserroles	48K
> aspnetusers	48K
> aspnetusertokens	16K
> bier	63K
Views	
Indexes	
Procedures	
Packages	