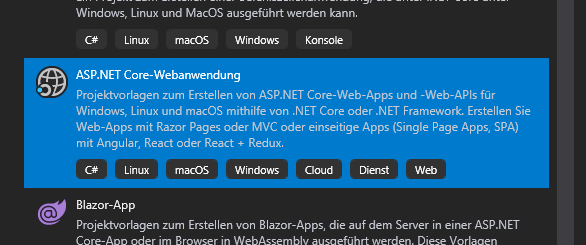
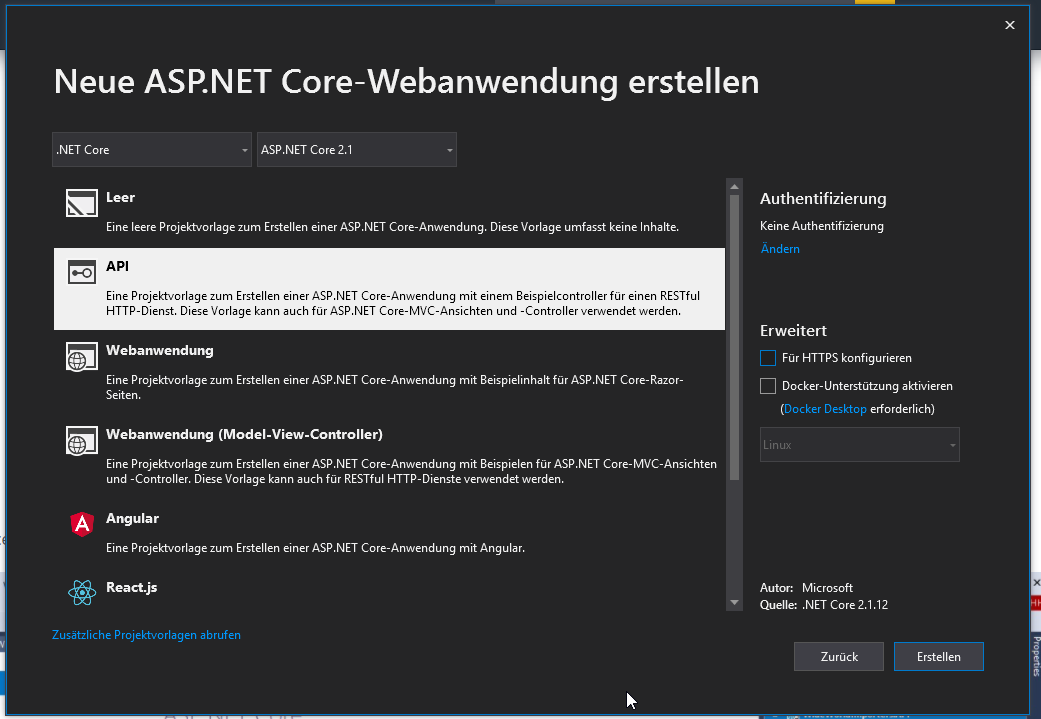
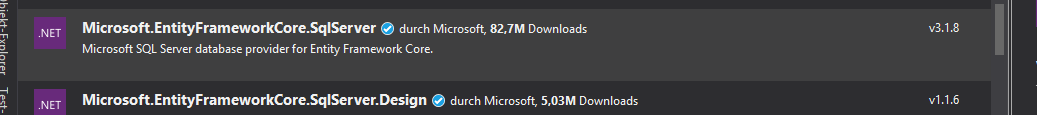
[https://www.codingame.com/playgrounds/35462/creating-web-api-in-asp-net-core-2-0/part-1---web-api](https://deref-gmx.net/mail/client/3euet5y3smY/dereferrer/?redirectUrl=https%3A%2F%2Fwww.codingame.com%2Fplaygrounds%2F35462%2Fcreating-web-api-in-asp-net-core-2-0%2Fpart-1---web-api)

ToDoWithLoginAPI

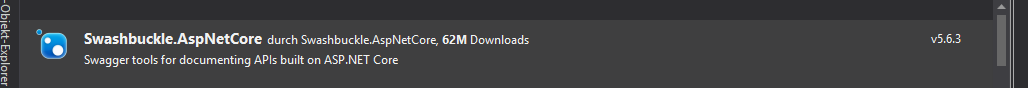




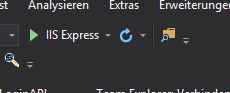
EntityFrameworkCore.SqlServer

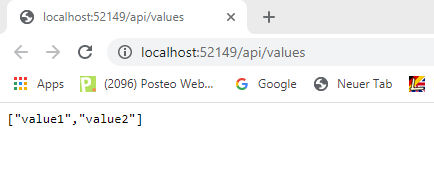


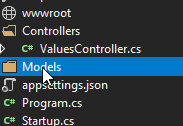
Swashbuckle.AspNetCore



<http://localhost:52149>







C:\Users\korn\source\repos\VueNew\ToDoWith LoginAPI2\ToDoWithLoginAPI\ToDoWithLoginAPI\Models\User.cs

public partial class User

{

    public User()

    {

    }

    public User(int id)

    {

        Id = id;

    }

    [Key]

    public int Id { get; set; }

    public string Name { get; set; }

    public string EmailAddress { get; set; }

    public string PasswordHash { get; set; }

}

C:\Users\korn\source\repos\VueNew\ToDoWith LoginAPI2\ToDoWithLoginAPI\ToDoWithLoginAPI\Models\ToDoItem.cs

public partial class TodoItem

{

    public TodoItem()

    {

    }

    public TodoItem(int id)

    {

        Id = id;

    }

    [Key]

    public int Id { get; set; }

    public string Name { get; set; }

    public bool IsCompleted { get; set; }

}

C:\Users\korn\source\repos\VueNew\ToDoWith LoginAPI2\ToDoWithLoginAPI\ToDoWithLoginAPI\Models\UsersConfiguration.cs

public class UsersConfiguration : IEntityTypeConfiguration<User>

{

    public void Configure(EntityTypeBuilder<User> builder)

    {

        // Set configuration for entity

        builder.ToTable("Users", "dbo");

        // Set key for entity

        builder.HasKey(p => p.Id);

        // Set configuration for columns

        builder.Property(p => p.Id).HasColumnType("int").IsRequired();

        builder.Property(p => p.Name).HasColumnType("nvarchar(50)");

        builder.Property(p => p.EmailAddress).HasColumnType("nvarchar(50)");

        builder.Property(p => p.PasswordHash).HasColumnType("nvarchar(50)");

        // Columns with default value

        builder

            .Property(p => p.Id)

            .HasColumnType("int")

            .IsRequired()

            .HasDefaultValueSql("NEXT VALUE FOR [Sequences].[Id]");

    }

}

C:\Users\korn\source\repos\VueNew\ToDoWith LoginAPI2\ToDoWithLoginAPI\ToDoWithLoginAPI\Models\ToDoItemsConfiguration.cs

    public class ToDoItemsConfiguration :  IEntityTypeConfiguration<TodoItem>

    {

        public void Configure(EntityTypeBuilder<TodoItem> builder)

    {

        // Set configuration for entity

        builder.ToTable("TodoItems", "dbo");

        // Set key for entity

        builder.HasKey(p => p.Id);

        // Set configuration for columns

        builder.Property(p => p.Id).HasColumnType("int").IsRequired();

            builder.Property(p => p.Name).HasColumnType("nvarchar(50)");

            builder.Property(p => p.IsCompleted).HasColumnType("bit");

        // Columns with default value

        builder

            .Property(p => p.Id)

            .HasColumnType("int")

            .IsRequired()

            .HasDefaultValueSql("NEXT VALUE FOR [Sequences].[Id]");

    }

}

C:\Users\korn\source\repos\VueNew\ToDoWith LoginAPI2\ToDoWithLoginAPI\ToDoWithLoginAPI\Models\ToDoWithLoginAPIContext.cs

public class ToDoWithLoginAPIContext : DbContext

{

    public ToDoWithLoginAPIContext(DbContextOptions<ToDoWithLoginAPIContext> options)

        : base(options)

    {

    }

    protected override void OnModelCreating(ModelBuilder modelBuilder)

    {

        // Apply configurations for entity

        modelBuilder

            .ApplyConfiguration(new UsersConfiguration());

        modelBuilder

            .ApplyConfiguration(new ToDoItemsConfiguration());

        base.OnModelCreating(modelBuilder);

    }

    public DbSet<User> Users { get; set; }

    public DbSet<TodoItem> TodoItems { get; set; }

}

C:\Users\korn\source\repos\VueNew\ToDoWith LoginAPI2\ToDoWithLoginAPI\ToDoWithLoginAPI\Startup.cs

public void ConfigureServices(IServiceCollection services)

{

    services.AddDbContext<ToDoWithLoginAPIContext>(options =>

    {

        options.UseSqlServer(Configuration["AppSettings:ConnectionString"]);

    });

    services.AddMvc().SetCompatibilityVersion(CompatibilityVersion.Version\_2\_1);

}

C:\Users\korn\source\repos\VueNew\ToDoWith LoginAPI2\ToDoWithLoginAPI\ToDoWithLoginAPI\appsettings.json

"AppSettings": {

  "ConnectionString": "server=(local);database=ToDoWith LoginAPI;integrated security=yes;"