

# Repository by Entity(ORM)



EntityFrameWork

DDD .

- Entity : ,
- Value Object : (ID) .
- Repository : , ,
- AggreGate() : , , )

```
<!-- ORM -->
<PackageReference Include="Microsoft.EntityFrameworkCore" Version="3.1.2" />
<PackageReference Include="Microsoft.EntityFrameworkCore.InMemory" Version="3.1.2" />
<PackageReference Include="Microsoft.EntityFrameworkCore.Sqlite" Version="3.1.2" />
<PackageReference Include="Pomelo.EntityFrameworkCore.MySql" Version="3.1.1" />
```

## Entity

```
namespace AkkaDotBootApi.Entity.User
{
    public class UserEntity
    {
        public int UserId { get; set; }

        public string Name { get; set; }

        public string Email { get; set; }
    }
}
```

Entity , POCO .



.

1.0 5 .

,

OOP ,

.

[https://ko.wikipedia.org/wiki/Plain\\_Old\\_Java\\_Object](https://ko.wikipedia.org/wiki/Plain_Old_Java_Object)

POCO , Entity () .

.Net core Fluent API Entity Repository .

- FluentAPI : <https://www.entityframeworktutorial.net/efcore/fluent-api-in-entity-framework-core.aspx>

## Repository

```
using AkkaDotBootApi.Config;
using AkkaDotBootApi.Entity.User;
using Microsoft.EntityFrameworkCore;

namespace AkkaDotBootApi.Repositories
{
    interface IUserRepository
    {
    }

    public class UserRepository : DbContext, IUserRepository
    {
        private string database = "webnori";

        protected DbSet<UserEntity> userEntities { get; set; }

        protected AppSettings appSettings { get; set; }

        public UserRepository(AppSettings _appSettings)
        {
            appSettings = _appSettings;
        }

        protected override void OnModelCreating(ModelBuilder modelBuilder)
        {
            modelBuilder
                .Entity<UserEntity>(eb =>
                {
                    eb.HasKey(b => b.UserId)
                        .HasName("PrimaryKey_UserId");

                    eb.ToTable("tb_user");
                });
        }

        protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
        {
            string dbOption = "Convert Zero Datetime=true;";
            string dbConnectionString = appSettings.DBConnectionMysql + $"database={database};" +
dbOption;

            optionsBuilder.UseMySQL(dbConnectionString);
            base.OnConfiguring(optionsBuilder);
        }
    }
}
```