

09. NativeQuery



Entity , Native

, Entity SQL

- ORM , ORM
- Dapper , Entity Native Query . ()
- (only ORM or only SQL) (ORM or NOT) Entity Framework .

Code Sample

```

//Entity
public class SomeLog
{
    [Key]
    public int no { get; set; }

    public string text { get; set; }

    public Datetime created_time { get; set; }
}

//Repository
public class LogRepository : DbContext
{
    private readonly AppSettings appSettings;

    public DbSet<SomeLog> someLogs { get; set; }

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

    protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
    {
        if (!optionsBuilder.IsConfigured)
        {
            string dbConnectionString = appSettings.DBConnectionString;
            //DB,
            optionsBuilder.UseNpgsql(dbConnectionString, o => o.SetPostgresVersion(8, 0));
        }
    }
}

// Service
public class LogSummaryService
{
    private ILogger _logger;
    private LogRepository _logRepository;

    public LogSummaryService(ILogger<LogSummaryService> logger,
        LogRepository logRepository)
    {
        _logger = logger;
        _logRepository = logRepository;
    }

    // SQL
    public async Task<List<SomeLog>> findDailySummaryList(SearchFilter searchFilter)
    {
        var query ="select .....;"
        var result = _logRepository.someLogs
            .FromSqlRaw(query).ToList();

        return result;
    }

    // ORM
    public async Task<List<SomeLog>> findDetailByFilter(SearchFilter searchFilter)
    {
        var item = _logRepository.someLogs.FirstOrDefault();
        return item;
    }
}

```

Key

```
protected override void OnModelCreating(ModelBuilder modelBuilder)
{
    modelBuilder
        .Entity<QueryStringResult>(eb =>
        {
            eb.HasNoKey();
        });
}
```

ORM Entity Key .

Native Query Key . Key .

```
var password = new MySqlParameter("password", passWord);
var result = await Native.FromSqlRaw(@"SELECT to_base64(SHA2(@password,512)) AS result;", password).
ToListAsync();
```

SQL Param , .

Param SQL (sql injection)

SQL

SQL SqlParameter .

:

- [ASP.NET Core](#)