

SecilStore

Backend Developer Code Case – Teknik Dökümantasyon

Hazırlayan: İpek Akdağ

Tarih: 15 Haziran 2025

Proje Adı: Dinamik Konfigürasyon Yönetim Sistemi

SecilStore

1. Proje Amacı

Projenin amacı web.config, app.config, appsettings.json gibi dosyalarda tutulan appkey'lerin ortak ve dinamik bir yapıyla erişilebilir olması ve deployment veya restart, recycle gerektirmeden güncellemelerin yapılabilir olmasıdır. Uygulama çalışırken konfigürasyon değerleri değiştirilebilmeli, sistem yeniden başlatılmadan bu değişiklikler uygulamaya yansıtılabilmelidir.

2. Genel Yapı

Proje üç bileşenden oluşur:

Proje Adı	Açıklama
ConfigurationApi	Web arayüzü (Konfigürasyonları listele, ekle, güncelle)
ConfigurationLibrary	Konfigürasyon okuma kütüphanesi (DLL)
ConfigurationTestApp	Konsol uygulaması (Kütüphane Testi)

3. Kütüphane Yapısı (ConfigurationLibrary)

Amaç:

Web, WCF, API gibi farklı uygulamalar tarafından kullanılabilen bir DLL olarak merkezi konfigürasyon verilerini okur.

Kullanım:

```
var reader = new ConfigurationReader(appName, connStr, interval);
string siteName = reader.GetValue<string>("SiteName");
Console.WriteLine("Site Adı: " + siteName);
```

Parametreler:

- **applicationName:** Uygulama adı (örneğin "SERVICE-A")
- **connectionString:** MSSQL (`string connStr = "Server=(localdb)\\MSSQLLocalDB;Database=DynamicConfigDb;Trusted_Connection=True;"`)
- **refreshInterval:** Yenileme aralığı (ConfigurationLibrary.cs - ms cinsinden)

Özellikler:

- Dapper ile MSSQL'den veri okuma
- Sadece IsActive = 1 kayıtları getirir
- MemoryCache ile hızlı erişim sağlar
- Belirli aralıklarla konfigürasyonları yeniler
- Tip dönüşümleri otomatik yapılır (string, int, bool, double)
- Dinamik yapılandırma: Yeni ayarlar eklemek için kod yazmaya gerek yok.
- CRUD işlemleri için web tabanlı Razor Pages arayüzü.
- GetValue<T>(string key) ile kolay erişim.
- Sadece aktif kayıtları gösteren yapı.
- İstemci taraflı arama (search) desteği.

4. Proje Yapısı

Solution

- ConfigurationLibrary // DLL olarak kullanılabilen yapı
 - ConfigurationReader.cs
 - Models/ConfigurationItem.cs
 - ConfigurationApi // Razor Pages arayüz
 - Pages/Index.cshtml
 - wwwroot/images/secil-logoblack.png
 - ConfigurationTestApp
-

5. Test Ortamı (ConfigurationTestApp)

- Konsol uygulamasıyla ConfigurationLibrary kütüphanesi test edildi
- SERVICE-A uygulaması için veriler başarıyla okundu
- Test edilen örnekler:
 - SiteName → string
 - IsBasketEnabled → bool
 - MaxItemCount → int

6. Veritabanı Yapısı (Configurations Tablosu)

Alan Adı	Veri Tipi	Açıklama
Id	int	Birincil anahtar
Name	string	Konfigürasyon anahtarı
Type	string	Veri tipi (string, int, ...)
Value	string	Değer
IsActive	bit	Aktif kayıt mı
ApplicationName	string	Hangi servise ait

7. Web Arayüz (ConfigurationApi)

- Listeleme, filtreleme, kayıt ekleme ve güncelleme yapılabilir
- Client-side filtreleme ile kullanıcı deneyimi artırıldı
- Kayıtlar anlık olarak değiştirilebilir

8. Karşılanan Gereksinimler

- .NET 8 ile geliştirilmiş kütüphane
- Her uygulama sadece kendi konfigürasyonlarına erişebiliyor
- Timer ile belli aralıklarla veri kontrolü sağlanıyor
- IsActive = 1 filtrelemesi yapılıyor
- T GetValue<T>(string key) metodu ile esnek kullanım
- Cache sayesinde performans ve offline toleransı sağlandı
- Web UI ile yönetim arayüzü hazırlandı

9. Sonuç

Proje temel gereksinimleri eksiksiz karşılamaktadır.

Yapı ölçeklenebilir şekilde tasarlanmış olup, birçok servise entegre edilebilir.

Geliştirme sürecinde MemoryCache, Dapper, Timer, Generic Method, Type Conversion, Layer Separation gibi konular başarıyla uygulanmıştır.

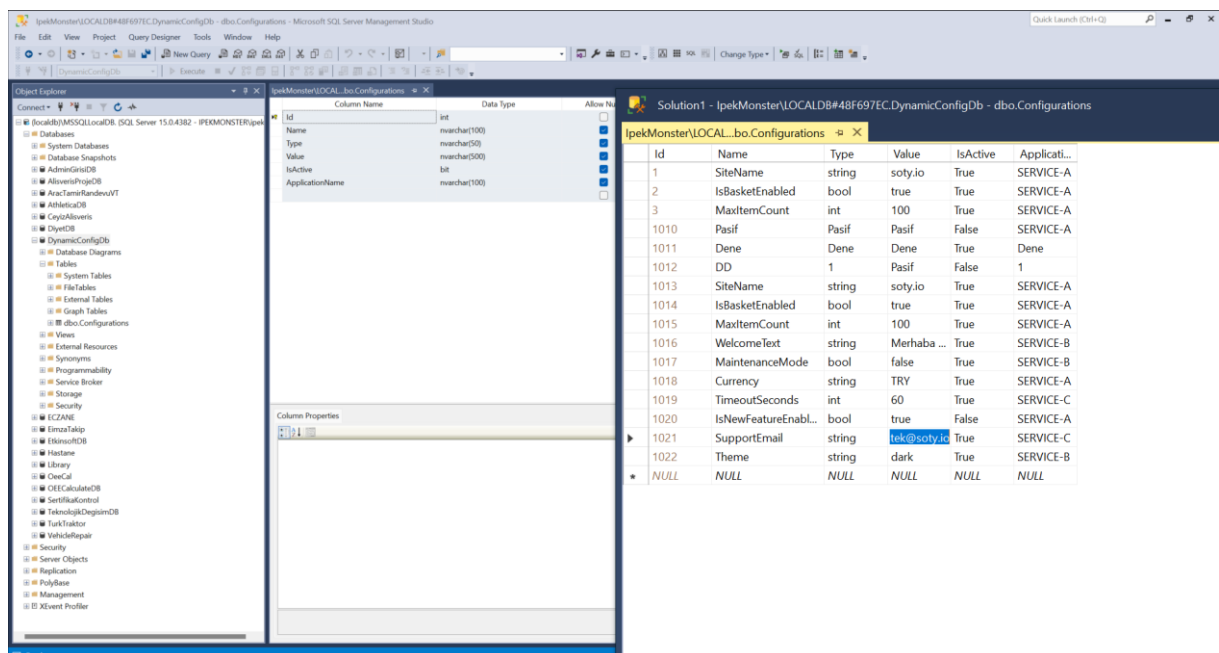
10. Kurulum- GitHub

- Kod deposu: (<https://github.com/ipekakdg/DynamicConfigurationProjectSecilStore>)
- DB Script:

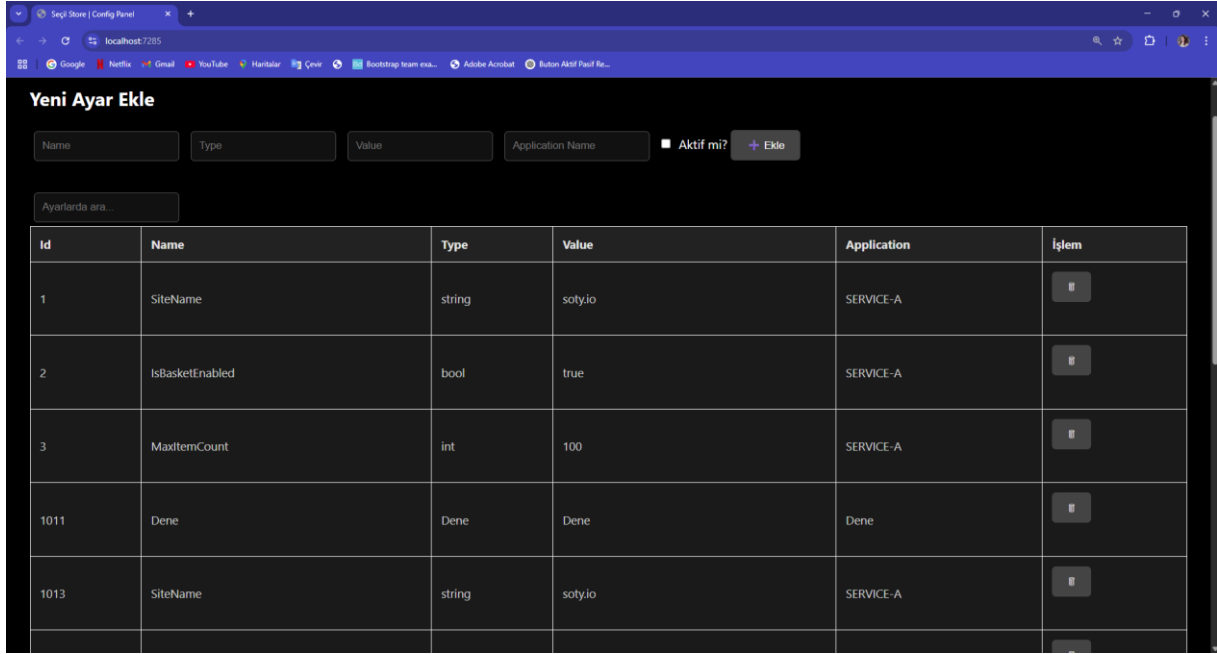
```
{  
  
CREATE TABLE Configurations (  
  
    Id INT IDENTITY(1,1) PRIMARY KEY,  
  
    Name NVARCHAR(100) NOT NULL,  
  
    Type NVARCHAR(50) NOT NULL,  
  
    Value NVARCHAR(500) NOT NULL,  
  
    IsActive BIT NOT NULL,  
  
    ApplicationName NVARCHAR(100) NOT NULL  
  
);  
  
GO  
}
```

11. Ekran görüntüleri

1. Veritabanı Görseli



2. Web Görşeli



3. Visual Studio Görşeli

