

# IOptions Pattern Nedir?

IOptions, .NET Core ile hayatımıza giren, strongly-typed configuration erişimi sağlayan bir pattern'dir.

ConfigurationManager.AppSettings["key"] uğraşlarından kurtaran yenilikçi yöntemdir. Tip güvenli, test edilebilir ve maintainable kod yazmayı kolaylaştıran bir adım.

## Üç Seçenek Vardır

### 1 IOptions<T>

- Singleton olarak register edilir
- Uygulama başlangıcında bir kez yüklenir
- Performans açısından en verimli
- Configuration değişikliklerini takip etmez

### 2 IOptionsSnapshot<T>

- Scoped olarak register edilir
- Her request'te yeniden yüklenir
- Configuration değişikliklerini yakalar
- Named options desteği vardır

### 3 IOptionsMonitor<T>

- Singleton olarak register edilir
- Real-time configuration değişikliklerini takip eder
- Change notification callback'leri destekler
- En esnek seçenek



# Örnek Kullanımım

```
"Maintenance": {  
  "Enabled": false,  
  "EndDateTime": "2025-08-01T15:30:00"  
}
```

AppSettings.json içine kullanacağımız bilgileri ekliyoruz.

```
public record MaintenanceOption  
{  
    public bool Enabled { get; init; }  
    public DateTime EndDateTime { get; init; }  
}
```

Aynı tip ve isimler ile option class oluşturuyoruz.

```
builder.Services.Configure<MaintenanceOption>(builder.Configuration.GetSection("Maintenance"));
```

Burada Program.cs içinde option class tanıtımını yaparız.

```
public class MaintenanceMiddleware  
{  
    private readonly RequestDelegate next;  
    private readonly MaintenanceOption options;  
    private readonly ILogger<MaintenanceMiddleware> logger;  
  
    public MaintenanceMiddleware(  
        RequestDelegate next,  
        IOptionSnapshot<MaintenanceOption> options,  
        ILogger<MaintenanceMiddleware> logger)  
    {  
        this.next = next;  
        this.options = options.Value;  
        this.logger = logger;  
    }  
  
    public async Task InvokeAsync(HttpContext context)  
    {  
        // Sadece bakım modu aktifse ve istek kendisi bakım sayfasına değilse  
        if (this.options.Enabled  
            && DateTime.Now < this.options.EndDateTime  
            && !context.Request.Path.StartsWithSegments("/Maintenance"))  
        {  
            this.logger.LogInformation(  
                "Bakım modu {EndDateTime} tarihine kadar aktif. {Method} isteği  
                this.options.EndDateTime,  
                context.Request.Method,  
                context.Request.Path,  
                context.Request.QueryString,  
                context.Connection.RemoteIpAddress);  
  
            context.Response.Redirect("/Maintenance");  
            return;  
        }  
  
        await this.next(context);  
    }  
}
```

Burada örnek bir middleware bulunmakta. Amacımız proje bakım moduna girdiği zaman bakım ekranına yönlendirmek. Proje ilk açılırken MaintenanceOption bir kez doldurulur ve burada her istekte `options` doldurulan değer ile kontrol edilir.

`IOptionSnapshot` kullanılırsa her seferinde AppSettings.json kontrol edilir.

Bu sayede projelerde her seferinde configuration nesnesine ulaşarak oradan get işlemi ile tamamen geliştiricinin isim vermesine güvenerek yapılan işlemler yerine eşleşen objelerin doldurulması güvenli geliştirmeyi sağlar.

Yeni projelerde bütün appsettings erişimlerinde kullanacağım bir yöntem olacağını düşünüyorum.