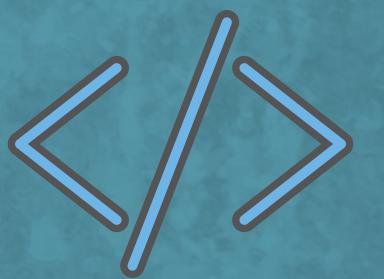


MIDDLEWARE

Clientdan gelen request neticesinde response dönene kadar olacak süreçlerin yapılacak işlemlerin sorumluluğunu üstlenecek olan Middleware dediğimiz ara yazılımlardır. Middleware tekrar eden işlemleri üstlenen, amaca yönelik görevleri gerçekleştiren çözümlerden biridir. Genellikle tüm route rotalarında yapılacak işlemleri tek elden yönetmek için kullanılmaktadır. Pipeline içerisine yeni bir aşama ekliyoruz gibi düşünebiliriz.

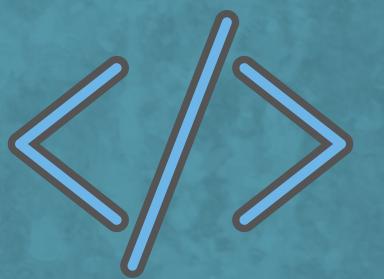


Büşra Altun

MIDDLEWARE

Birçok kişiyi davet ettiğiniz bir partiye ev sahipliği yaptığınızı hayal edin. Hepsi farklı diller konuşuyor ve hiçbirini diğerini anlamıyor. Tüm dilleri anlayan ve konuşan tek kişi sensin. Başarılı bir partiye sahip olmak ve iyi bir ev sahibi olmak için herkesin birbiriyle iletişim kurabilmesi için çevirmen rolünü üstlenmeniz gereklidir.

Ara yazılım tam olarak böyledir - birden fazla sistemin farklı platformlarda birbirleriyle iletişim kurmasını sağlar. Bazıları, daha fazla sayıda uygulamayı içeren Kurumsal düzeyde, bazıları ise daha küçük ölçekte yararlanır. Ara yazılım önemlidir çünkü bu uygulamalar arasında sinerji ve entegrasyonu mümkün kılar.

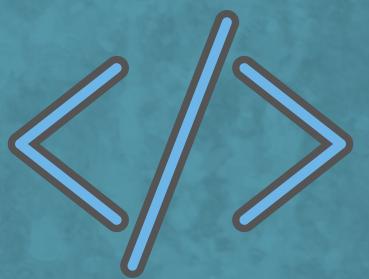


Büşra Altın

MIDDLEWARE

Uygulamaların eşzamanlı (anında yanıt) veya eşzamansız (gecikmeli yanıt) iletişim kurmasını sağlar. Bu arada Request e müdahale etmek isteriz çünkü kullanıcıya gecikmeli bir yanıt döndürmek isteyebiliriz mesela.

Aradan geçen bilgilerin şifrelenmesini, şifresinin çözülmesini sağlayarak bilgilerin güvenliğini sağlayabilir.



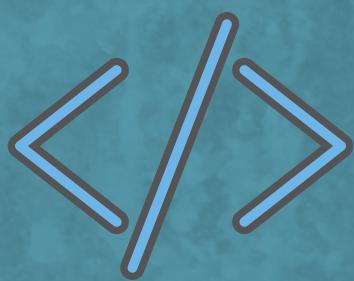
Büşra Altın



BİR YAZILIMCI HANGİ DURUMLARDA MIDDLEWARE E İHTİYAÇ DUYAR

Geliştiricilerin uygulamaları daha verimli bir şekilde oluşturmasına yardımcı olur. Uygulamalar, veriler ve kullanıcılar arasında connective tissue gibi davranışır. Geliştiricinin gerekli donanımı kullanması için çekirdek ile iletişim kurmasını kolaylaştırır.

Connective tissue : Bir şeyi başka bir şeye veya birçok şeyi topluca birbirine tutturmak için kullanılır

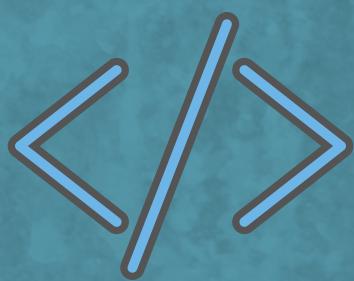


Büşra Altın



BİR YAZILIMCI HANGİ DURUMLARDA MIDDLEWARE E İHTİYAÇ DUYAR

Geliştiriciler, birden fazla araç, dil ve çerçeve ile uğraşıyor. Ve daha az zamanda ve daha düşük maliyetle daha fazlasını yapması gerekebiliyor veya istenebiliyor. Veritabanı işlemleri, uygulama sunucuları arasındaki işlemler, ileti kontrolleri, kimlik doğrulama gibi amaçlar doğrultusunda ara yazılımlar (middleware) oluşturulabilmektedir.



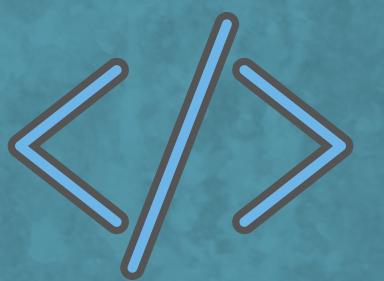
Büşra Altın



MIDDLEWARE E NEDEN İHTİYAÇ DUYULUR

Çoklu bulut ve kapsayıcı ortamlara sahip kuruluşlar için, ara yazılım, uygulamaları uygun ölçekte geliştirmeyi ve çalıştırmayı uygun maliyetli hale getirebilir.

Connective tissue : Bir şeyi başka bir şeye veya birçok şeyi topluca birbirine tutturmak için kullanılır

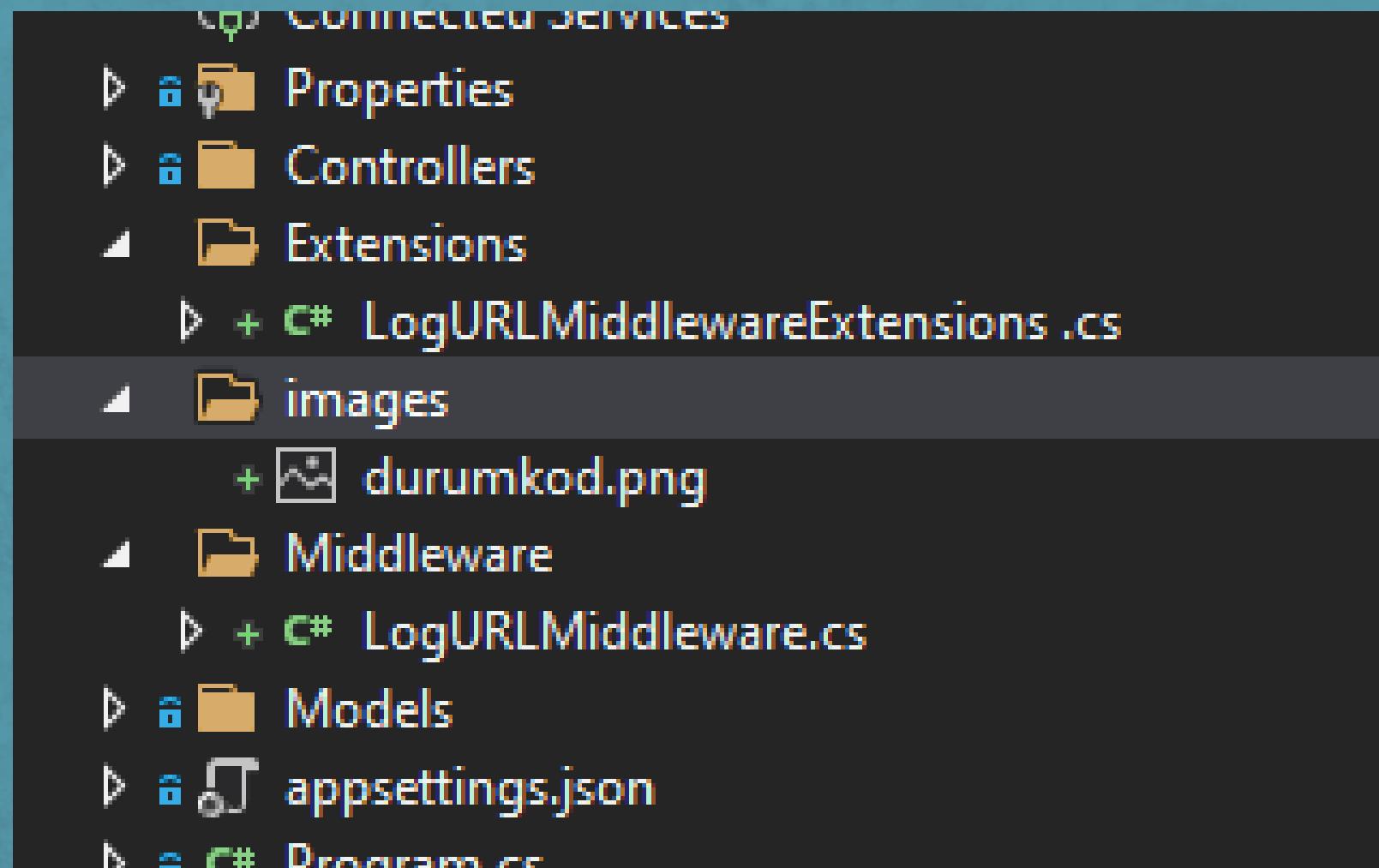


Büşra Altın

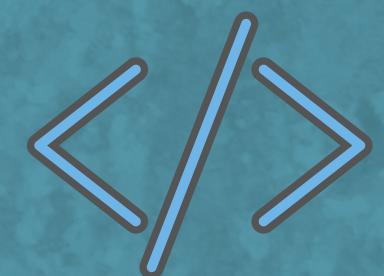
1

Middleware İçin Senaryolar

Dizine göz atma, projemizde kullanıcıların belirli bir dizindeki bir dizin listesini, dosyaları görmesine olanak tanıma.



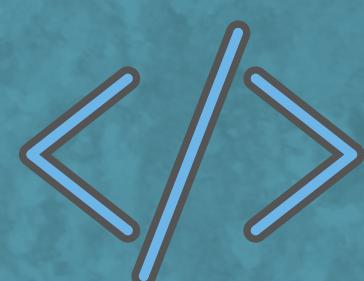
| Name | Size | Last Modified |
|--------------|---------|---------------------------|
| durumkod.png | 111.532 | 4.06.2021 14:03:35 +00:00 |



Büşra Altın

1

```
app.UseDirectoryBrowser(new DirectoryBrowserOptions
{
    FileProvider = new PhysicalFileProvider(Path.Combine(Directory.GetCurrentDirectory(), "images")),
    RequestPath = "/images"
});
```



2

Middleware İçin Senaryolar

Her request geldiğine "Hoşgeldin" şeklinde bir cümle karşılaşın

Bunun için bir extension ve bir middleware oluşturup gerekli çağrıları yapıp startup da
ayağa kaldırıdım.

```
Content root path: C:\Users\Desktop\BusraAltun_Ode
Hoşgeldin, birazdan istedigin sayfaya kavusacaksın
İstedigim sayfay gösterdiğim için teşekkür ederim
```



Büşra Altun

2

```
Middleware.cs    WelcomeMiddleware.cs  ✘  Startup.cs
Dapper_Micro_ORM_Web_API
Dapper_Micro_ORM_Web_API.Middleware.WelcomeMiddleware
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace Dapper_Micro_ORM_Web_API.Middleware
{
    2 başvuru
    public class WelcomeMiddleware
    {
        RequestDelegate _next; //using Microsoft.AspNetCore.Http; kullanıldı

        0 başvuru
        public WelcomeMiddleware(RequestDelegate next)
        {
            _next = next;
        }

        0 başvuru
        public async Task Invoke ( HttpContext httpContext )
        {
            //Custom operasyon

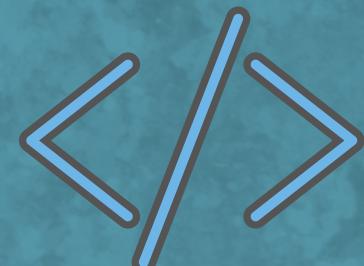
            Console.WriteLine("Hosgeldin, birazdan istedigin sayfaya kavusacaksın");
            await _next.Invoke(httpContext); //Bir sonrakini Invoke() ediyoruz ki kısa devre olmasın
            Console.WriteLine("Istedigim sayfayı gösterdigin için teşekkür ederim");
        }
    }
}
```



Büşra Altın

2

```
1  using Dapper_Micro_ORM_Web_API.Middleware;
2  using Microsoft.AspNetCore.Builder;
3  using System;
4  using System.Collections.Generic;
5  using System.Linq;
6  using System.Threading.Tasks;
7
8  namespace Dapper_Micro_ORM_Web_API.Extensions
9  {
10    public static class ExtensionForMiddleware
11    {
12      public static IApplicationBuilder UseHello(this IApplicationBuilder applicationBuilder)
13      {
14        //Middleware ler genelde IApplicationBuilder döndürüler
15        return applicationBuilder.UseMiddleware<WelcomeMiddleware>();
16      }
17    }
}
```



Büşra Altun

3

Middleware İçin Senaryolar

Kategori query string varsa bunu ekrana yazdırılm yoksa " no query string" yazdırılm

```
app.Run(async (context) =>
{
    if (context.Items["value"] != null)
    {
        var value = context.Items["values"].ToString();
        await context.Response.WriteAsync($"category : {value}");
    }
    else
    {
        await context.Response.WriteAsync("no query string");
    }
});
```

