

# Yazılım Mimarileri ve Tasarım Desenleri Üzerine

C#, .NET Core, Asp.NET Core, LINQ, Web Service, jQuery, T-SQL, SQL Server, Ajax, OOP, Design Patterns, Design Principles, Entity

Framework Core, Angular 9/10, Node.js, Express.js, MongoDB, NoSQL, Azure(Storage, Functions), RabbitMQ, Docker, Redis, SignalR,

IdentityServer4, OData

## GTM TEKNOLOJİ

GTM Teknoloji - Supermicro Türkiye Resmi Distribütörü - Server, S  
GPU Server

GTM Teknoloji

Asp.NET  
.NET Core  
.NET Core 3.0  
.NET Core 3.1  
Asp.NET Core  
Asp.NET Core MVC  
Asp.NET Core 2  
Asp.NET Core 2.1  
Asp.NET Core 2.2  
Asp.NET Core 3.0  
Asp.NET Core 3.1  
Asp.NET MVC  
Asp.NET MVC 5  
MVC 4.0  
C Sharp – C#  
C# 5.0  
C# 6.0  
C# 7.0  
C# 7.1  
C# 7.2  
C# 8.0  
C# 9.0  
Design Patterns  
Design Principles  
LINQ  
Entity Framework  
Entity Framework Core  
Diğer  
Action Script  
Akıllı Telefonlar  
E-Book  
Flash

Google Teknolojileri  
Adsense  
Php  
Güvenlik  
Hata Çözümleri  
HTML5  
İpuçları  
PhpBB  
Şiir Tadında  
Oyun  
Programlar  
SEO(Arama Motoru Optimizasyonu)  
Sosyal Ağ  
Web Tasarım  
WordPress  
Genel  
Güncel  
Videolar  
Angular Framework  
AngularJS  
Angular 2  
Angular 4  
Angular 5  
Angular 6  
Angular 7  
Angular 8  
Angular 9  
Javascript  
TypeScript  
Jquery  
KnockoutJS  
Ajax  
Veritabanı  
SQL Server  
SQL Server 2016  
XML  
JSON  
Java  
Win32  
Görüntü İşleme  
Accord.NET  
AForge.Net  
EmguCV  
Yazı Dizileri  
IdentityServer4 Yazı Serisi  
Asp.NET Core – SignalR Yazı Serisi  
Azure Functions Yazı Serisi  
Azure Storage Yazı Dizisi  
Asp.NET Core Web API – OData Yazı Dizisi  
Redis Yazı Dizisi  
.NET Core & Docker Yazı Dizisi  
RabbitMQ Yazı Dizisi  
Asp.NET Core Identity – Yazı Dizisi  
C#'ta Gösterici(Pointer) Yazı Dizisi  
Angular 4 Yazı Dizisi  
Node.js  
ECMAScript 6  
Express.js  
Pug  
NoSQL  
MongoDB

RabbitMQ  
Docker  
Redis  
OData  
Microservices  
SignalR  
Azure  
IdentityServer4  
gRPC

TAKİP ET:



SONRAKİ YAZI  
C# 7.0 – Deconstruct Metodu

ASP.NET CORE 2 / ASP.NET CORE 2 MVC /  
ASP.NET CORE 2.1 / ASP.NET CORE 2.2

3

DAHA FAZLA



ÖNCEKİ YAZI  
C# 8.0 – Default Implementations In  
Interfaces

Derinlemesine yazılım eğitimleri için kanalımı takip edebilirsiniz...

Kod yazmak sanattır...



Bilge Adam  
Software Trainer & Consultant

## Asp.NET Core'de In-Memory Cache

YAZAR: GENÇAY · 21 MAYIS 2019

Beğen 0 Tweetle Kaydet Paylaşır ?

### MSI NVIDIA GeForce GTX 1050 TI Garantili

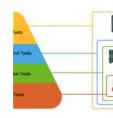
₺5.299

Bilgisayar &gt; Bilgisayar Bileşenleri &gt; Kartları &gt; MSI NVIDIA GeForce GTX 1050 TI Gaming X 4G 4GB 128 bit GDDR5 DX12 PCI-E 3.0 Ekran Kartı (MAG105XTI-GV-N4GD5)

Alışverişe başla



Asp.NET Core'da ki Exception'ları Otomatik Olarak Yönetin



.NET Core – Unit Test Nedir? Nasıl Yapılır? Derinlemesine İnceleyelim



Onion Architecture'da CQRS + MediatR Pattern + AutoMapper + Fluent Validation



CQRS Pattern Nedir? MediatR Kütüphanesi ile Nasıl Uygulanır?



Nedir Bu Onion Architecture? Tam Teferruatlı İnceleyelim

Asp.NET Core 5.0 –

## SAYFALAR

- Arşiv
- Bana Ulaşım
- C# Algoritma Soruları ve Çözümleri
- C# Eğitim Videoları
- Şahsüm Adına
- SQL Server ve T-SQL Eğitim Videoları
- Yazı Dizileri

## REKLAM

Merhaba,

Bir uygulamanın performansını önemli ölçüde artırbilecek en mühim yapılanmalardan biriside önbelleğe(Cache) alma sistemidir. Son kullanıcıya gösterilen veriler arasından nadiren güncellenen veya geniş aralıklarla tazelenen verileri her istek neticesinde veritabanı üzerinden elde ederek kullanıcuya sunmaktadır, ilgili verileri varsa periyodik sürelerde uygun bir aralıkta yahut tarafınızca belirlenmiş bir zaman sınırında önbelleğe alarak böylece orjinal veriyi tekrar elde etmek yerine daha hızlı sonuçlar üretebilirsiniz.

Asp.NET Core, modüler olarak içerisinde birkaç farklı önbellek yapılanmasını desteklemektedir. Bizler bu içeriğimizde, bu önbellek



#### YARDIMLAŞMAK !!!

Bir insanın yardım etme hissiyatı, diğerinin acizliği ya da ezikliğinden değildir. Bilakis onu kendinden çok insan sıfatıyla görmesindendir. Çünkü, insana hizmet etmek insana yakışır. Bu alanda veya başka bir alanda, benim ve diğer yardımcı insanların paylaşımımlarına lütfen acizliğiniz ve ezikliğinizle yaklaşmayın. İzin istemek, benim hükmüm altına girmeniz anlamına gelmemektedir.

#### KATEGORİLER

■ .NET Core (37)

■ .NET Core 3.0 (15)

yapılanlıklarından en basit olanı In-Memory Cache yapılanmasını inceleyeceğiz.

In-Memory Cache, önbellekleme sürecinde web uygulamasının barındırıldığı sunucunun belleğini kullanmaktadır ve "IMemoryCache" arayüzü aracılığıyla uygulamada çalışmaktadır.

```
public interface IMemoryCache : IDisposable
{
    ICacheEntry CreateEntry(object key);
    void Remove(object key);
    bool TryGetValue(object key, out object value);
}
```

Yukarıda "IMemoryCache" arayüzünün içeriğini incelerseniz eğer "CreateEntry", "Remove" ve "TryGetValue" metodları mevcuttur. Bu metodların sırasıyla işlevselliklerini incelersek eğer;

#### ■ CreateEntry

Memory Cache'de bir alan oluşturur ve içersine değer atamamızı sağlar.

#### ■ Remove

Memory Cache'de oluşturulan alanı temizlemizi sağlar.

#### ■ TryGetValue

Memory Cache'de belirtilen alanda varsa değeri getirir.

Asp.NET Core uygulamalarında In-Memory Cache özelliğini kullanabilmek için ilgili yapının modülünün uygulamaya service olarak eklenmesi gerekmektedir.

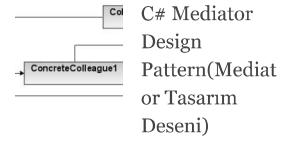
```
public class Startup
{
    public Startup(IConfiguration configuration)
    {
        Configuration = configuration;
    }
    public IConfiguration Configuration { get; }

    public void ConfigureServices(IServiceCollection services)
    {
        services.AddMvc();
        services.AddMemoryCache();
    }

    public void Configure(IApplicationBuilder app, IHostingEnvironment env)
    {
        app.UseMvc();
    }
}
```



SqlTableDependency  
Kütüphanesi ile Veritabanındaki Değişikleri Anlık Yakalama



C# Mediator Design Pattern(Mediator Tasarım Deseni)



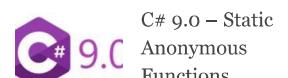
Deadlines ve Cancellation ile Güvenilir gRPC Servisleri Geliştirmek



Entity Framework Core ile In-Memory'de Çalışmak



IdentityServer4 Yayı Serisi #23  
IdentityServer4 Entity Framework ile Konfigürasyon Ayarlarını Veritabanına Taşıma



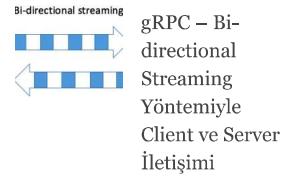
C# 9.0 - Static Anonymous Functions



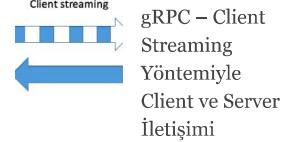
.NET Core 5.0 - Retry Pattern, Timeout ve Circuit Breaker | Polly



gRPC - File Streaming Nasıl Gerçekleştirilir ?



gRPC - Bi-directional Streaming Yöntemiyle Client ve Server İletişimi



gRPC - Client Streaming Yöntemiyle Client ve Server İletişimi

[.NET Core 3.1 \(8\)](#)[Accord.NET \(1\)](#)[Action Script \(1\)](#)[Adsense \(1\)](#)[AForge.Net \(4\)](#)[Ajax \(14\)](#)[Akıllı Telefonlar \(1\)](#)[Angular \(13\)](#)[Angular 4 \(44\)](#)[Angular 5 \(12\)](#)[Angular 6 \(10\)](#)[Angular 7 \(7\)](#)[Angular 8 \(5\)](#)[Angular 9 \(3\)](#)[AngularJS \(12\)](#)[Asp.NET \(61\)](#)[Asp.NET Core \(28\)](#)[Asp.NET Core 2 \(42\)](#)[Asp.NET Core 2 MVC \(45\)](#)[Asp.NET Core 2.1 \(31\)](#)[Asp.NET Core 2.2 \(30\)](#)[Asp.NET Core 3.0 \(34\)](#)[Asp.NET Core 3.1 \(21\)](#)[Asp.NET Core 5.0 \(3\)](#)[Asp.NET MVC \(111\)](#)[Asp.NET MVC 5 \(32\)](#)[Azure \(31\)](#)[C Sharp – C# \(251\)](#)[C# 5.0 \(31\)](#)[C# 6.0 \(27\)](#)[C# 7.0 \(8\)](#)[C# 7.1 \(1\)](#)

Bu işlemden sonra Memory Cache'i kullanabilmek için ilgili controllerda IMemoryCache arayüzüne inject etmemiz yeterlidir.

 Server streaming

gRPC – Server Streaming  
Yöntemiyle Client ve Server İletişimi

 Unary

gRPC – Unary  
Yöntemiyle Client ve Server İletişimi



gRPC'de Client ve Server Uygulamaları, Proto Dosyası ve Servislerin Oluşturulması



gRPC Nedir?  
Ne Amaçla ve Nasıl Kullanılır?



Almanca İş Fırsatı  
5.000 € Maaş + Yüksek Prim. Başlangıç  
Yüksek Kazanç Fırsatı.  
Vetra Group

```
[Route("api/[controller]")]
[ApiController]
public class PersonelController : ControllerBase
{
    IMemoryCache _memoryCache;
    NorthwindContext _northwindContext;
    public PersonelController(IMemoryCache memoryCache, NorthwindContext northwindContext)
    {
        _memoryCache = memoryCache;
        _northwindContext = northwindContext;
    }
}
```

Artık Cache'lenecek verileri Memory'e atabiliriz.

```
[Route("api/[controller]")]
[ApiController]
public class PersonelController : ControllerBase
{
    IMemoryCache _memoryCache;
    NorthwindContext _northwindContext;
    public PersonelController(IMemoryCache memoryCache, NorthwindContext northwindContext)
    {
        _memoryCache = memoryCache;
        _northwindContext = northwindContext;
    }
    [HttpGet("[action]")]
    public ActionResult<IEnumerable<Personeller>> Get()
    {
        const string key = "personeller";

        if (!_memoryCache.TryGetValue(key, out object list))
            return Ok(list);
        var personnelList =
            _northwindContext.Personeller.Select(x => new
            {
                x.Adi,
                x.SoyAdi,
                x.Unvan
            }).ToList();
        _memoryCache.Set(key, personnelList, new MemoryCacheEntryOptions
        {
            AbsoluteExpiration =

```

- » C# 7.2 (5)
- » C# 8.0 (9)
- » C# 9.0 (7)
- » Design Pattern (24)
- » Design Principles (5)
- » Diğer (33)
- » Docker (20)
- » ECMAScript 6 (8)
- » Eğitimler (1)
- » EmguCV (6)
- » Entity Framework (13)
- » Entity Framework Core (10)
- » Express.js (11)
- » Flash (1)
- » Genel (73)
- » Google Teknolojileri (5)
- » Görüntü İşleme (10)
- » gRPC (8)
- » Güncel (29)
- » Güvenlik (1)
- » Hata Çözümleri (37)
- » HTML5 (5)
- » IdentityServer4 (24)
- » İpuçları (2)
- » Java (3)
- » Javascript (48)
- » Jquery (65)
- » JSON (14)
- » KnockoutJS (7)
- » LINQ (18)
- » Microservices (8)
- » MongoDB (13)

```
        DateTime.Now.AddSeconds(20),
        Priority =
        CacheItemPriority.Normal
    });
    return Ok(personnelList);
}
}
```

Yukarıdaki kod bloğunu incelerseniz eğer “Set” metodunun 3. parametresinde “MemoryCacheEntryOptions” tipinden bir nesne istenmektedir. Bu nesne belleğe atılan verinin temel konfigürasyonunu sağlamakta, ne kadar süre bellekte tutulacağını ve memory şıstiğinde cache objelerini hangi önceliğe göre sileceğini vs. bilgilerini tutmaktadır. Bunların dışında; yukarıda yapılan işlem neticesinde ilk request ile orjinal veri elde edilecek ve Memory Cache’e “20” saniyeliğine atılacaktır. Bu “20” saniye boyunca yapılan tüm requestler ilgili source’e gitmeksızın cacheden karşılaşacaktır. Expire süresi dolan datalar ilgili cacheden temizlenecektir.

Tüm bu işlemlerin dışında Cache data set edildiği gibi remove’da edilebilmektedir.

```
_memoryCache.Remove(key);
```

Evet görüldüğü üzere bu kadar basit...

İlgilenenlerin faydalaması dileğiyle...  
Sonraki yazılarımda görüşmek üzere...  
İyi çalışmalar...

i x

## Almanca İş Fırsatı

5.000 ₺ Maaş + Yüksek Prim. Başarılı Bir Kariyer ve Yüksek Kazanç Fırsatı.

VetraCo Group

Hemen Başvurun

Beğen 0
Tweetle
Kaydet
Paylaşır
?

Etiketler: [Asp.NET Core](#) [Asp.NET Core Cache](#)  
[Asp.NET Core In-Memory Cache](#) [IMemoryCache](#) [In-Memory Cache](#)  
[MemoryCacheEntryOptions](#)

**BUNLAR DA HOŞUNUZA GİDEBİLİR...**





- » MVC 4.0 (50)
- » Node.js (47)
- » NoSQL (9)
- » OData (9)
- » Oyun (3)
- » Php (2)
- » PhpBB (1)
- » Programlar (9)
- » Pug (2)
- » RabbitMQ (13)
- » Redis (14)
- » SEO(Arama Motoru Optimizasyonu) (16)
- » SignalR (11)
- » Şiir Tadında (8)
- » Sosyal Ağ (1)
- » SQL Server (22)
- » SQL Server 2016 (21)
- » TypeScript (4)
- » Veritabanı (54)
- » Videolar (33)
- » Visual Studio (1)
- » WCF (1)
- » Web API (7)
- » Web Tasarım (12)
- » Win32 (5)
- » WordPress (11)
- » WPF (1)
- » XML (13)
- » Yazı Dizeleri (9)

.NET Core – Asp.NET Core Asp.NET Core  
 FluentValidation İdentity – Yazı Identity –  
 ile Dizisi Politika Bazlı  
 Validasyon 10 AĞUSTOS Kimlik  
 Kontrolü 2019 Doğrulama –  
 03 MART 2020 XVI  
 08 KASIM 2019

### 3 CEVAPLAR

» Yorumlar 3 » Referans Bildirimi 0



**borahan** Ⓜ 14 Nisan 2020, 19:48

Merhaba hocam parametreli gönderimlerde işe yaramaz sanırım 😞 mesala bunu sanırım database üzerinde dil olaylarında kullanıyorlar galiba, belki ülkelerde kullanılır, makale için teşekkürler

Cevapla



**murat** Ⓜ 03 Şubat 2021, 11:37

Güzel bir anlatım olmuş elinize sağlık. Şöyledir bir durum var ama ben mi yanlış anladım bilemiyorum. Bu Cache örnek olarak attığımız hostun kendi üzerinde oluşturulup tutuluyor. Client tarafında bir cacleme işlemi olmuyor. Doğru mudur?

Cevapla



**Gençay** Ⓜ 03 Şubat 2021, 21:08

Merhaba,

Evet, doğrudur.

Cevapla

Reklam 

**Get All Mobirise Themes and Extensions for the T**

1500+ Awesome Blocks. 27 Fresh Bootstrap4/AMP Then Save Huge!

[Mobirise Website Builder](#)

### BİR CEVAP YAZIN

#### Yorum

İsim \*

E-posta \*

Internet sitesi





CAPTCHA (Şahıs Denetim) Kodu

\*

Yorum gönder

## Supermicro Türkiye

GTM Teknoloji - Supermicro Türkiye Resmi  
Distribütörü - Server, Storage & GPU Server  
GTM Teknoloji

### ARŞİVLER

■ Nisan 2021 (2)

■ Mart 2021 (4)

■ Şubat 2021 (4)

■ Ocak 2021 (10)

■ Aralık 2020 (12)

■ Kasım 2020 (19)

■ Ekim 2020 (14)

■ Eylül 2020 (10)

■ Ağustos 2020 (20)

■ Temmuz 2020 (12)

■ Haziran 2020 (5)

■ Mayıs 2020 (16)

■ Nisan 2020 (26)

■ Mart 2020 (22)

■ Şubat 2020 (14)

■ Ocak 2020 (2)

■ Aralık 2019 (4)

■ Kasım 2019 (5)

■ Ekim 2019 (6)

■ Eylül 2019 (4)

■ Ağustos 2019 (15)

■ Temmuz 2019 (2)

■ Haziran 2019 (7)

■ Mayıs 2019 (6)

---

■ Nisan 2019 (7)

---

■ Mart 2019 (6)

---

■ Şubat 2019 (8)

---

■ Ocak 2019 (7)

---

■ Aralık 2018 (6)

---

■ Kasım 2018 (2)

---

■ Ekim 2018 (6)

---

■ Eylül 2018 (13)

---

■ Ağustos 2018 (14)

---

■ Temmuz 2018 (27)

---

■ Haziran 2018 (10)

---

■ Mayıs 2018 (8)

---

■ Nisan 2018 (18)

---

■ Mart 2018 (34)

---

■ Şubat 2018 (1)

---

■ Ocak 2018 (1)

---

■ Aralık 2017 (1)

---

■ Kasım 2017 (1)

---

■ Ekim 2017 (1)

---

■ Eylül 2017 (3)

---

■ Ağustos 2017 (5)

---

■ Temmuz 2017 (8)

---

■ Haziran 2017 (7)

---

■ Mayıs 2017 (13)

---

■ Nisan 2017 (15)

---

■ Mart 2017 (7)

---

■ Şubat 2017 (12)

---

■ Ocak 2017 (3)

---

■ Aralık 2016 (10)

---

■ Kasım 2016 (5)

---

■ Ekim 2016 (12)

---

■ Eylül 2016 (5)

---

■ AĞUSTOS 2016 (6)

---

■ TEMMUZ 2016 (12)

---

■ HAZİRAN 2016 (15)

---

■ MAYIS 2016 (8)

---

■ NİSAN 2016 (31)

---

■ MART 2016 (19)

---

■ ŞUBAT 2016 (7)

---

■ OCAK 2016 (11)

---

■ ARALIK 2015 (5)

---

■ KASIM 2015 (7)

---

■ EKİM 2015 (5)

---

■ EYLÜL 2015 (14)

---

■ AĞUSTOS 2015 (10)

---

■ TEMMUZ 2015 (5)

---

■ HAZİRAN 2015 (1)

---

■ MAYIS 2015 (1)

---

■ NİSAN 2015 (1)

---

■ MART 2015 (3)

---

■ ŞUBAT 2015 (1)

---

■ OCAK 2015 (2)

---

■ ARALIK 2014 (3)

---

■ KASIM 2014 (2)

---

■ EKİM 2014 (2)

---

■ EYLÜL 2014 (3)

---

■ AĞUSTOS 2014 (8)

---

■ TEMMUZ 2014 (11)

---

■ HAZİRAN 2014 (10)

---

■ MAYIS 2014 (8)

---

■ NİSAN 2014 (4)

---

■ MART 2014 (15)

---

■ ŞUBAT 2014 (8)

---

■ OCAK 2014 (11)

---

■ Aralık 2013 (3)

---

■ Kasım 2013 (8)

---

■ Ekim 2013 (6)

---

■ Eylül 2013 (20)

---

■ Ağustos 2013 (12)

---

■ Temmuz 2013 (16)

---

■ Haziran 2013 (11)

---

■ Mayıs 2013 (8)

---

■ Nisan 2013 (5)

---

■ Mart 2013 (13)

---

■ Şubat 2013 (13)

---

■ Ocak 2013 (15)

---

■ Aralık 2012 (16)

---

■ Kasım 2012 (16)

---

■ Ekim 2012 (2)

---

■ Eylül 2012 (4)

---

■ Ağustos 2012 (5)

---

■ Temmuz 2012 (4)

---

■ Haziran 2012 (7)

---

■ Mayıs 2012 (20)

---

■ Nisan 2012 (13)

---

■ Mart 2012 (90)



Bu sitedeki tüm içerik @Gençay Yıldız'a aittir. Sitenin tüm hakkı gizlidir ve bana aittir ayrıyetten çalıntı ve kaynaksız alıntı yapılmadığı taktirde hakkımlı hepinize helaldır.

Geliştirici: - Designed with the Hueman theme