

1. Kullanılan araçlar ve teknolojiler

- VS2013
- SQLExpress 2008
- Entity Framework
- Mvc
- PagedList.Mvc
- Generic Repository Pattern
- HttpRuntime.Cache

2. Açıklamalar

- Mvc template projesinden yola çıkılarak Products isimli bir sayfa eklenmiştir. Headerdan ilgili linke tıklayarak ilgili işlemler yapılabilir. Template ile gelen gereksiz sayfalar silinmemiştir.
- Data üretimi için aşağıdaki script kullanılmıştır ve 20 kadar kayıt el ile girilmiştir.

```
DECLARE @cnt INT = 0;
WHILE @cnt < 100000
BEGIN
    Insert ProductModel(Name,LastUpdatedTime)VALUES('xxxxxxxxxx', '2015-12-20')
    SET @cnt = @cnt + 1;
END;
PRINT 'Done';
GO
```

- Cache işlemleri için kurulum gerektiren app katmanındaki cache yapıları yerine HttpRuntime.Cache tercih edilmiştir. CacheHelper classı içerisinde cache işlemleri yapılmıştır.
- Belirli aralıklarla cachein güncellenmesi işlemi için global.asax içerisindeki aşağıdaki kod satırı kullanılmıştır.

```
UpdateCache();
Task.Run(async () =>
{
    while (true)
    {
        await Task.Delay(new TimeSpan(0, 0, 10)); //belli aralıklarla cachei güncelleyecek bir thread oluşturuldu
        UpdateCache();
    }
});
```

- Herhangi bir ürün eklendiğinde yada güncellendiğinde GenericRepository içerisindeki Save methodunda cache de güncellenmektedir.

```
3 references
public void Save(string cacheKey)
{
    dbContext.SaveChanges();
    IEnumerable<T> updatedObjects = this.SelectAll();
    CacheHelper.UpdateCache(cacheKey, updatedObjects);
}
```

- Sadece güncellenen datanın cache üzerinde update edilmesi denenmiş ancak 100K data için cachei topluca okumanın ve yazmanın(where koşulu bulunmadan) daha performanslı olduğu görülmüştür. Dolayısıyla cachei toptan güncelleme yoluna gidilmiştir.

3. İyileştirme

Şu an repository save edilirken cache güncellenerek işlem merkezileştirildi, bu cache IIS bazında olduğu için bir kişi değiştirdiğinde tüm kullanıcılar için değişir. Ancak cache değişimi clientların browserı refresh olmadıkça etkilemez. Bunun yapılabilmesi için; IHttpAsyncHandler ve IHttpHandlerdan türetilmiş iki handlerdan faydalanılabildi. AsyncHandler ile hook atılır(beginprocessrequest), diğer handler ile hook atmış tüm kullanıcıların browserı trigger edilebilirdi yada signalR gibi apileri kullanma yoluna gidilebilirdi.

4. Ekran Görüntüleri

- Ana ekran

Trendyol CodeCase

Home

Products

Products

[Create New](#)

Find by name:

Name	LastUpdatedTime	
aaatestest	21.12.2015 23:54:08	Edit
andarsdbcdw	21.12.2015 21:19:00	Edit
bobb	12.12.2015 00:00:00	Edit
dfo	21.12.2015 21:19:00	Edit
dhjghj	21.12.2015 21:19:00	Edit
eeer	21.12.2015 21:19:00	Edit
fc	21.12.2015 21:19:00	Edit
fdhgfd	21.12.2015 21:19:00	Edit
ghdf	21.12.2015 21:19:00	Edit
qweqwe	21.12.2015 21:19:00	Edit

Page 1 of 10003

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) ... [»](#) [»»](#)

- Arama işlemi

PRODUCTS

[Create New](#)

Find by name:

Name	LastUpdated Time	
aaatestest	21.12.2015 23:54:08	Edit
test	12.12.2015 00:00:00	Edit
test22222222	21.12.2015 23:38:27	Edit

Page 1 of 1

[1](#)

- Yeni ürün yaratma

Create

Product Create

Name

[Back to List](#)

- Ürün güncelleme(ürünlerin sağ tarafındaki edit linkine tıklanarak)

Edit

Product

Name

[Back to List](#)