**IMemoryCache**

متد های پراپرتی های این اینترفیس به شرح زیر است:

Set(string key, object value);

این متد با استفاده از شناسه یکتا به نام کلید، مقدار مورد نظر را در حافظه رم نگهداری می کند.

Get(string key);

این متد با استفاده از کلیدی که دریافت می کند به دنبال مقدار Cache شده می گردد.

TryGetValue(string key, out object data);

این متد برای چک کردن وجود کلید مورد نظر در حافظه رم می باشد، در صورت وجود کلید، مقدار data با مقداری که شناسه آن، کلید دریافت شده توسط متد می باشد، پر می شود.

GetOrCreate(string key);

متد به دنبال کلید دریافت شده میگردد در صورت چنین کلیدی در حافظه رم موجود باشد آن را بر می گرداند، در غیر این صورت ابتدا آن را ایجاد کرده و مقدار آن را بر میگرداند.

Remove(object key);

این متد یک آیتم از حافظه Cache حذف می کند.

برای مدیریت بهتر فضای حافظه Cache باید اطلاعات مربوط به محدود سازی Cache های خود را داخل کلاسی به نام MemoryCacheEntryOptions به متد Set هنگام ساخت Cache بدهیم، که در ادامه به این تنظیمات می پردازیم.

این اطلاعات توسط اینترفیس ICacheEntry در اینترفیس IMemoryCache و متدی به نام CreateEntry() پر میشود. که مقدار دهی این تنظیمات توسط پارامتر سوم متد Set() (همانطور که پیش تر گفته شد) نیز صورت می گیرد.

DateTimeOffset? AbsoluteExpiration { get; set; }

این پراپرتی ساعت دقیق از بین رفتن آیتم Cache را در خود نگه می دارد.به عبارت دیگر اینکه آیتم Cache در چه ساعتی از بین برود.

TimeSpan? SlidingExpiration { get; set; }

این پراپرتی نشان می دهد که آیتم Cache چه قدر بعد از آخرین فراخوانی آیتم از حافظه Cache از بین برود.

CacheItemPriority Priority { get; set; }

اولویت آیتم Cache را در حافظه Cache مشخص می کند. یعنی اگر نیاز به حذف خودکار یکی از آیتم ها باشد، عملیات حذف با توجه به اولویت آن صورت می گیرد.

long? Size { get; set; }

سایز مقدار Cache شده را بر می گرداند.

object Value { get; set; }

مقدار Cache شده را بر می گرداند.

object Key { get; }

کلید مقدار Cache شده را بر می گرداند.

IList<IChangeToken> ExpirationTokens { get; }

IChangeToken اینترفیسی است که مشخص می کند که داده مورد نظر تغییر کرده است یا خیر. به عبارت دیگر تاریخچه تغییر داده را در خود نکه می دارد.در بحث IMemoryCache نیز تاریخچه تغییرات آیتم های Cache را در خود نگه می دارد.

IList<PostEvictionCallbackRegistration> PostEvictionCallbacks { get; }

لیست کارها به علاوه وضعیت آن ها را نگه میدارد. منظور از لیست کارها، کارهایی است که هنگام ازبین رفتن آیتم Cache اجرا می شود.

public PostEvictionDelegate EvictionCallback { get; set; }

کاری را هنگام از بین رفتن آیتم Cache اجرا می شود.