C#/.NET编程进阶

[集合和数据结构](https://docs.microsoft.com/zh-cn/dotnet/standard/collections/selecting-a-collection-class)

[线程安全集合](https://docs.microsoft.com/zh-cn/dotnet/standard/collections/thread-safe/)

[为多线程处理同步数据](https://docs.microsoft.com/zh-cn/dotnet/standard/threading/synchronizing-data-for-multithreading)

[托管线程处理的最佳做法](https://docs.microsoft.com/zh-cn/dotnet/standard/threading/managed-threading-best-practices)

[线程处理对象和功能-同步基元概述](https://docs.microsoft.com/zh-cn/dotnet/standard/threading/overview-of-synchronization-primitives)

[.NET 中的并行处理、并发和异步编程](https://docs.microsoft.com/zh-cn/dotnet/standard/parallel-processing-and-concurrency)

[lock 语句（C# 参考）](https://docs.microsoft.com/zh-cn/dotnet/csharp/language-reference/keywords/lock-statement)

[.NET多线程并发与同步](https://www.jianshu.com/p/ea79ba42bcc0)

[多线程中的锁 volatile、Interlocked、ReaderWriterLockSlim](https://cloud.tencent.com/developer/article/1131776)

[C#中的多线程 - 高级多线程](https://blog.gkarch.com/threading/part4.html)

ConcurrentBag<T>、BlockingCollection<T>

相关参考：

[C#/.NET Little Wonders: ConcurrentBag and BlockingCollection](http://geekswithblogs.net/BlackRabbitCoder/archive/2011/03/03/c.net-little-wonders-concurrentbag-and-blockingcollection.aspx)

List<T> is a collection designed to use in single thread applications.

ConcurrentBag<T> is a subtype of ConcurrentCollection<T> designed to simplify using collections in multi-thread environments. If you use ConcurrentCollection you will not have to lock your collection to prevent corruption by other threads. You can insert or take data from your collection with no need to write special locking codes.

BlockingCollection<T> is designed to get rid of the requirement of checking if new data is available in shared collection between threads. if there is new data inserted to shared collection than your consumer thread will awake immediatily. So you do not have to check if new data is available for consumer thread in certain time intervals typically in a while loop.

Asp.Net Core Mvc

//add a action filter

services.AddMvc(opts=> {

opts.Filters.Add(typeof(ModelStateFeatureFilter));

})