

Rate limiting in ASP.NET

C# Setup.cs

```
var builder = WebApplication.CreateBuilder(args);
builder.Services.AddRateLimiter(options =>
{
    /* Policy configuration here */
});
```

```
var app = builder.Build();
app.UseRateLimiter();
app.Run();
```

C# Usage.cs

```
// Global configuration
builder.Services.AddRateLimiter(options =>
{
    options.GlobalLimiter = /* Implementation here */
});

// Specific endpoints (also works with MVC Controllers/Endpoints)
app.MapGet("/", () => Results.Ok("Hello world"))
    .RequireRateLimiting(Policy);
```



timdeschryver.dev/bits

Fixed window limiter

Sliding window limiter

Token bucket limiter

Concurrency limiter

Rate limiting in ASP.NET: Fixed window limiter

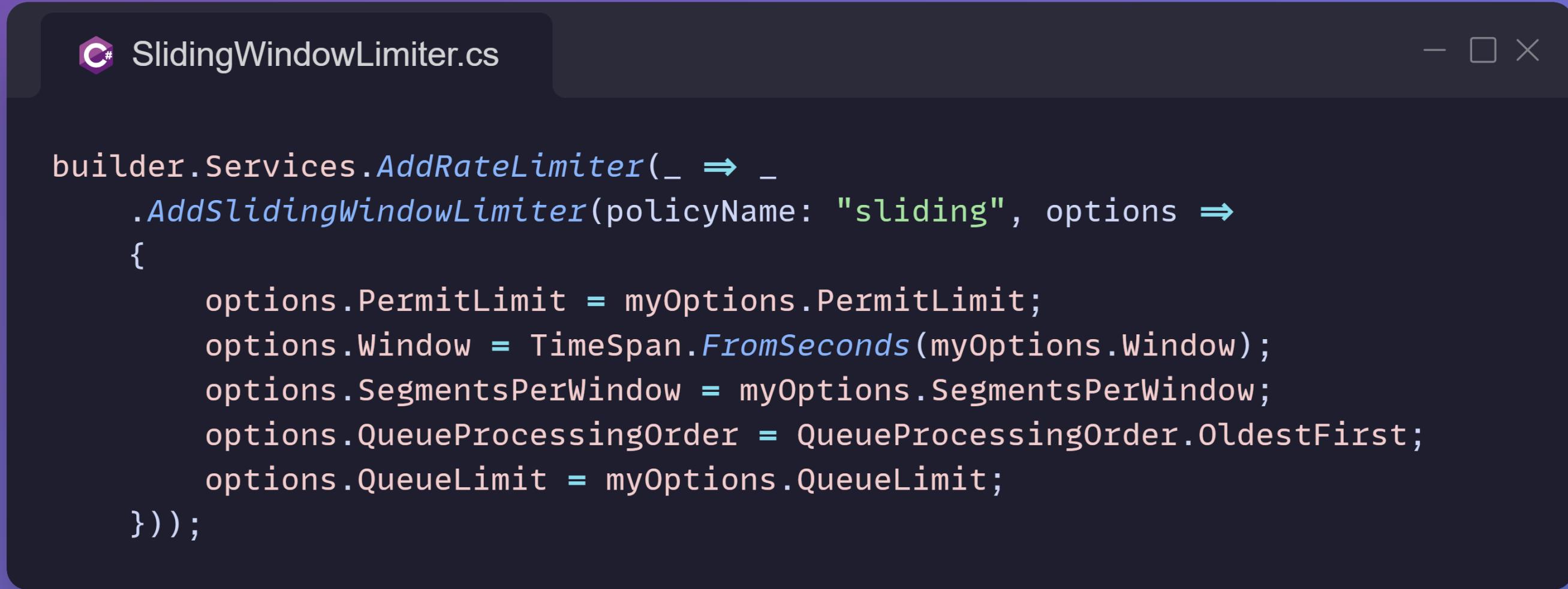


The image shows a dark-themed code editor window with a title bar that includes a C# icon and the file name "FixedWindowLimiter.cs". The main area of the editor contains the following C# code:

```
builder.Services.AddRateLimiter(_ => _  
    .AddFixedWindowLimiter(policyName: "fixed", options =>  
    {  
        options.PermitsLimit = myOptions.PermitsLimit;  
        options.Window = TimeSpan.FromSeconds(myOptions.Window);  
        options.QueueProcessingOrder = QueueProcessingOrder.OldestFirst;  
        options.QueueLimit = myOptions.QueueLimit;  
    }) );
```

The **AddFixedWindowLimiter** method uses a fixed time window to limit requests. When the time window expires, a new time window starts and the request limit is reset.

Rate limiting in ASP.NET: Sliding window limiter

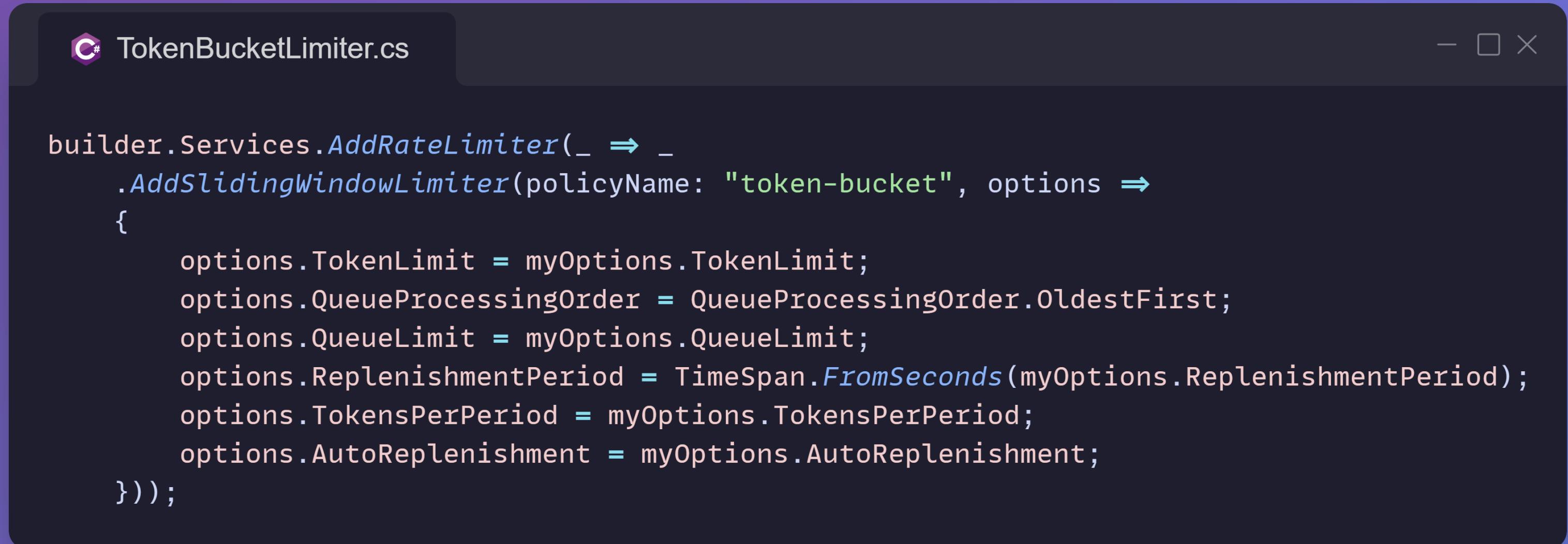


The image shows a dark-themed code editor window with a title bar "C# SlidingWindowLimiter.cs". The main area contains C# code for configuring a sliding window rate limiter:

```
builder.Services.AddRateLimiter(_ => _  
    .AddSlidingWindowLimiter(policyName: "sliding", options =>  
    {  
        options.PermitsLimit = myOptions.PermitsLimit;  
        options.Window = TimeSpan.FromSeconds(myOptions.Window);  
        options.SegmentsPerWindow = myOptions.SegmentsPerWindow;  
        options.QueueProcessingOrder = QueueProcessingOrder.OldestFirst;  
        options.QueueLimit = myOptions.QueueLimit;  
    }) );
```

The sliding window is similar to the fixed window limiter but adds segments per window. The window slides one segment each segment interval. The segment interval is (window time)/(segments per window).

Rate limiting in ASP.NET: Token bucket limiter



The image shows a code editor window with a dark theme. The title bar says "TokenBucketLimiter.cs". The code in the editor is as follows:

```
builder.Services.AddRateLimiter(_ => _  
    .AddSlidingWindowLimiter(policyName: "token-bucket", options =>  
    {  
        options.TokenLimit = myOptions.TokenLimit;  
        options.QueueProcessingOrder = QueueProcessingOrder.OldestFirst;  
        options.QueueLimit = myOptions.QueueLimit;  
        options.ReplenishmentPeriod = TimeSpan.FromSeconds(myOptions.ReplenishmentPeriod);  
        options.TokensPerPeriod = myOptions.TokensPerPeriod;  
        options.AutoReplenishment = myOptions.AutoReplenishment;  
    }) );
```

The token bucket limiter is similar to the sliding window limiter, but rather than adding back the requests taken from the expired segment, a fixed number of tokens are added each replenishment period. The tokens added each segment can't increase the available tokens to a number higher than the token bucket limit.

Rate limiting in ASP.NET: Concurrency limiter

C# ConcurrencyLimiter.cs

```
builder.Services.AddRateLimiter(_ => _  
    .AddConcurrencyLimiter(policyName: concurrencyPolicy, options =>  
    {  
        options.PermitsLimit = myOptions.PermitsLimit;  
        options.QueueProcessingOrder = QueueProcessingOrder.OldestFirst;  
        options.QueueLimit = myOptions.QueueLimit;  
    }) );
```

The concurrency limiter limits the number of concurrent requests. Each request reduces the concurrency limit by one. When a request completes, the limit is increased by one. Unlike the other requests limiters that limit the total number of requests for a specified period, the concurrency limiter limits only the number of concurrent requests and doesn't cap the number of requests in a time period.