

Новое в ASP.NET Core

Петров Сергей

SKAI

Метрики аутентификации и авторизации

- Authentication:
 - Authenticated request duration
 - Challenge count
 - Forbid count
 - Sign in count
 - Sign out count
- Authorization:
 - Count of requests requiring authorization

Метрики ASP.NET Core Identity

- aspnetcore.identity.user.create.duration
- aspnetcore.identity.user.update.duration
- aspnetcore.identity.user.delete.duration
- aspnetcore.identity.user.check_password_attempts
- aspnetcore.identity.user.generated_tokens
- aspnetcore.identity.user.verify_token_attempts
- aspnetcore.identity.sign_in.authenticate.duration
- aspnetcore.identity.sign_in.check_password_attempts
- aspnetcore.identity.sign_in.sign_ins
- aspnetcore.identity.sign_in.sign_outs
- aspnetcore.identity.sign_in.two_factor_clients_remembered
- aspnetcore.identity.sign_in.two_factor_clients_forgotten

401, 403 вместо редиректов

- Эндпоинты, использующие cookie аутентификацию теперь по умолчанию возвращают 401 или 403 статус коды вместо редиректа на страницу логина или access denied URI

- Чтобы вернуть предыдущее поведение:

```
builder.Services.AddAuthentication()
    .AddCookie(options =>
{
    options.Events.OnRedirectToLogin = context =>
    {
        context.Response.Redirect(context.RedirectUri);
        return Task.CompletedTask;
    };
    options.Events.OnRedirectToAccessDenied = context =>
    {
        context.Response.Redirect(context.RedirectUri);
        return Task.CompletedTask;
    };
});
```

Exception handler middleware

- Добавлено опция SuppressDiagnosticsCallback если не нужны диагностические логи. По умолчанию включена.
- Чтобы вернуть предыдущее поведение:

```
app.UseExceptionHandler(new ExceptionHandlerOptions
{
    SuppressDiagnosticsCallback = context => false;
});
```

Top-level statement и тесты

- Раньше необходимо было использовать:

public partial class Program, сейчас это не обязательно

```
0 references
40 Using public partial class Program { } to make the generated Program class public is no longer
41 required in ASP.NET Core apps. See https://aka.ms/aspnetcore-warnings/ASP0027 for more details.
42 (ASP0027)
43 Quick Fix... (⌘.)
44 public partial class Program {}
```

Автоматическая очистка из MemoryPool

- MemoryPool используемый kestrel (или IIS, Http.sys) стал автоматически освобождать блоки памяти при бездействии или малой нагрузке.
- Работает автоматически
- Добавлена метрика для отслеживания:

`Microsoft.AspNetCore.MemoryPool`

```
public class MyBackgroundService : BackgroundService {
    private readonly MemoryPool<byte> _memoryPool;
    public MyBackgroundService(IMemoryPoolFactory<byte> factory) {
        _memoryPool = factory.Create();
    }
    protected override async Task ExecuteAsync(CancellationToken stoppingToken) {
        while (!stoppingToken.IsCancellationRequested) {
            try {
                await Task.Delay(20, stoppingToken);
                var rented = _memoryPool.Rent(100); // do work that needs memory
                rented.Dispose();
            }
            catch (OperationCanceledException) {
                return;
            }
        }
    }
}
```

Minimal APIs

При использовании [FromForm] атрибута пустые строки конвертируются в null

```
var builder = WebApplication.CreateBuilder(args);
var app = builder.Build();
app.MapPost("/todo", ([FromForm] Todo todo) => TypedResults.Ok(todo));
app.Run();

public class Todo
{
    public int Id { get; set; }
    public DateOnly? DueDate { get; set; } // Empty strings map to `null`
    public string Title { get; set; }
    public bool IsCompleted { get; set; }
}
```

Minimal APIs validation

Поддержка валидации:

- Query
- Header
- Request body

Minimal APIs validation

Поддержка валидации:

- Query
- Header
- Request body

Кастомизация через:

- IValidatableObject
- [Validation]

Minimal APIs validation

Добавить:

```
builder.Services.AddValidation()
```

Отключить для эндпоинта:

```
app.MapPost("/products",
    ([EvenNumber(ErrorMessage = "Product ID must be even")] int productId,
     [Required] string name)
        => TypedResults.Ok(productId)).DisableValidation();
```

Minimal APIs validation

- Валидации можно использовать и в record

```
public record Product(  
    [Required] string Name,  
    [Range(1, 1000)] int Quantity);
```

- Они будут применяться автоматически

```
app.MapPost("/products", (Product product) =>  
{  
    // Endpoint logic here  
    return TypedResults.Ok(product);  
});
```

Minimal APIs Server-Sent Events (SSE)

Поток событий через одно http соединение:

передача `SseItem<T>` объектов используя `TypedResults.ServerSentEvents`

```
app.MapGet("/json-item", (CancellationToken cancellationToken) =>
{
    async IAsyncEnumerable<HeartRateRecord> GetHeartRate(
        [EnumeratorCancellation] CancellationToken cancellationToken)
    {
        while (!cancellationToken.IsCancellationRequested)
        {
            var heartRate = Random.Shared.Next(60, 100);
            yield return HeartRateRecord.Create(heartRate);
            await Task.Delay(2000, cancellationToken);
        }
    }
    return TypedResults.ServerSentEvents(GetHeartRate(cancellationToken),
        eventType: "heartRate");
});
```

РЕСУРСЫ

- [learn.microsoft.com – новое в dotnet 10](https://learn.microsoft.com/ru-ru/dotnet/what-is-dotnet-10)

