APM dla .NET — dokumentacja komponentu

Wersja: .NET (ASP.NET Core/API/Worker), integracja z Kibana APM  
*Data: 2025-08-21 05:56*

# 1) Cel komponentu

* End-to-end observability dla usług .NET: czasy odpowiedzi, błędy, zależności (SQL/HTTP), dystrybucja opóźnień.
* Wsparcie decyzji operacyjnych: SLO/alerty, analiza regresji po deployu, korelacja z logami.

# 2) Zakres i funkcje

* Automatyczny tracing dla: ASP.NET Core, HttpClient, SQL Server (ADO.NET), gRPC; metryki runtime (.NET GC, wątki).
* Rozszerzenia: Entity Framework Core, stack traces, wyłapywanie wyjątków.
* Distributed tracing (W3C traceparent), korelacja z logami (trace.id / transaction.id).
* Konfigurowalne: próbkowanie, prywatność (capture\_body, sanitize\_field\_names).

# 3) Wymagania

* .NET 6/7/8 (x64), aplikacja: ASP.NET Core/API/Worker/IIS/Windows Service.
* APM Server/Kibana (adresy, token/API key, certyfikaty TLS).
* Uprawnienia do ustawienia zmiennych środowiskowych lub edycji konfiguracji aplikacji.

# 4) Instalacja — dwa warianty

## 4A) Wariant „NuGet + kod” (najprostszy przy re-deploy)

1) Dodaj pakiety (co najmniej jeden): `Elastic.Apm.AspNetCore` lub agregujący `Elastic.Apm.NetCoreAll`.

Opcjonalnie: `Elastic.Apm.EntityFrameworkCore`, `Elastic.Apm.SqlClient`, `Elastic.Apm.SerilogEnricher`.

2) Program.cs (ASP.NET Core 6+/7+/8+):

using Elastic.Apm.AspNetCore;  
  
var builder = WebApplication.CreateBuilder(args);  
var app = builder.Build();  
  
// Wczyta konfigurację z appsettings.json i/lub zmiennych środowiskowych:  
app.UseElasticApm(builder.Configuration);  
  
app.Run();

3) Konfiguracja w appsettings.json (przykład):

{  
 "ElasticApm": {  
 "ServerUrls": "https://<APM\_SERVER>/",  
 "SecretToken": "<token>",  
 "ServiceName": "MyApi",  
 "Environment": "prod",  
 "TransactionSampleRate": 0.3,  
 "CaptureBody": "errors", // off | errors | transactions | all  
 "SanitizeFieldNames": "password,token,Authorization,Set-Cookie"  
 }  
}

4) Alternatywnie (bez appsettings) przez zmienne środowiskowe:

ELASTIC\_APM\_SERVER\_URLS=https://<APM\_SERVER>/  
ELASTIC\_APM\_SECRET\_TOKEN=<token> # lub ELASTIC\_APM\_API\_KEY=<key>  
ELASTIC\_APM\_SERVICE\_NAME=MyApi  
ELASTIC\_APM\_ENVIRONMENT=prod  
ELASTIC\_APM\_TRANSACTION\_SAMPLE\_RATE=0.3  
ELASTIC\_APM\_CAPTURE\_BODY=errors  
ELASTIC\_APM\_SANITIZE\_FIELD\_NAMES=password,token,Authorization,Set-Cookie  
ELASTIC\_APM\_LOG\_LEVEL=Info # Debug do diagnostyki

## 4B) Wariant „Auto-instrumentation (profiler)” — bez zmian w kodzie

Dobre dla IIS/Windows Service i miejsc, gdzie nie chcesz przebudowywać aplikacji.

1) Rozpakuj binaria agenta (profiler) obok aplikacji i/lub do wspólnej ścieżki (offline: przygotuj ZIP wcześniej).

2) Ustaw zmienne środowiskowe na poziomie procesu/usługi (IIS App Pool/Windows Service):

ELASTIC\_APM\_SERVER\_URLS=https://<APM\_SERVER>/  
ELASTIC\_APM\_SECRET\_TOKEN=<token>  
ELASTIC\_APM\_SERVICE\_NAME=MyService  
ELASTIC\_APM\_ENVIRONMENT=prod  
ELASTIC\_APM\_LOG\_LEVEL=Info

3) Upewnij się, że profiler jest ładowany (CORECLR\_PROFILER, CORECLR\_PROFILER\_PATH skonfigurowane przez instalator/agenta).

4) Zrestartuj usługę/IIS. Sprawdź logi agenta i APM w Kibanie.

Uwaga: jeśli używasz własnego CA, dodaj zaufanie CA do cert store hosta; nie wyłączaj weryfikacji TLS w produkcji.

# 5) Kibana — gdzie patrzeć i co wnioskować

* APM → Services → Twoja usługa: Latency (avg/p95/p99), Throughput, Failed transaction rate, Errors, Dependencies, Time spent by span type, Instances latency distribution.
* APM → Traces: powiązania wielu usług (distributed tracing).
* APM → Service maps: topologia i gorące ścieżki.
* Analytics → Dashboards/Lens: własne wykresy na indeksach apm-\*.

# 6) KQL — szybkie filtry (Discover / Lens na apm-\*)

Błędy w usłudze:

service.name : "MyApi" and processor.event : "error"

Powolne transakcje (>1 s):

service.name : "MyApi" and processor.event : "transaction" and transaction.duration.us >= 1000000

Czas w DB (span SQL):

service.name : "MyApi" and processor.event : "span" and span.type : "db" and span.subtype : "mssql"

# 7) Alerty i SLO — wzorce

* Latency threshold: p95 > 500 ms przez 5 min (service.name=MyApi, transaction.type=request).
* Error rate threshold: > 5% w 5–10 min, z grupowaniem po transaction.name.
* Anomalie (ML): alert na odchylenia latencji (jeśli włączone).
* SLO: SLI „request < 500 ms i bez błędu”, cel 99.5%; monitoruj burndown.

# 8) Korelacja z logami (Serilog / Filebeat)

Dodaj korelację trace.id/transaction.id do logów:

using Serilog;  
using Elastic.Apm.SerilogEnricher;  
  
Log.Logger = new LoggerConfiguration()  
 .Enrich.WithElasticApmCorrelationInfo()  
 .WriteTo.Console(outputTemplate: "{Timestamp:HH:mm:ss} {Level:u3} {Message} trace={TraceId} tx={TransactionId}{NewLine}")  
 .CreateLogger();

W Filebeat prześlij te pola (lub sparsuj z message), aby w Kibanie przechodzić z loga do śladu APM.

# 9) Własne transakcje i spany (instrumentacja custom)

using Elastic.Apm;  
using Elastic.Apm.Api;  
  
Agent.Tracer.CaptureTransaction("Order Checkout", ApiConstants.TypeRequest, t =>  
{  
 t.CaptureSpan("Call Payments API", "external", s =>   
 {  
 // ... kod zewnętrznego wywołania ...  
 });  
});

Normalizuj nazwy transakcji (np. „GET /orders/:id”), aby ograniczać kardynalność.

# 10) Dobre praktyki

* Prywatność: CaptureBody=errors (lub off) + SanitizeFieldNames dla sekretów.
* Sampling: zacznij od 0.3; dopasuj do RPS i krytyczności ścieżek.
* Nazewnictwo: ServiceName, Environment (prod/stage/dev), spójne transaction.name.
* Health checks: wyklucz /health z metryk.
* Release annotations: ustaw service.version, by wykrywać regresje po deployu.
* Zależności: weryfikuj span.type=db/http/queue; doinstaluj brakujące instrumentacje.

# 11) Troubleshooting

* 401/403 do APM Server: sprawdź token/API key i endpoint ServerUrls.
* TLS: zaufanie dla CA po stronie hosta; nie wyłączaj weryfikacji w prod.
* Brak danych: czy UseElasticApm jest wywołany (wariant 4A)? Czy profiler się wstrzyknął (wariant 4B)?
* Brak SQL spans: użyj wspieranego providera ADO.NET i upewnij się, że diagnostyka jest włączona.
* Wysoki narzut: obniż TransactionSampleRate, ogranicz CaptureBody, wyłącz zbędne instrumentacje.
* Logi agenta: ELASTIC\_APM\_LOG\_LEVEL=Debug — przeanalizuj output.