Elliot One • 2nd
AI-Powered SaaS Builder | Founder @ XANT & Monoversity | Entrepreneur
11h •

Follow

Health checks are essential for building resilient and reliable systems.

They play a crucial role in microservices and distributed architectures.

Health checks enable you to monitor the health of key application components, including databases, external APIs, caches, and other services.

ASP .NET Core makes it easy to implement health checks:
➜ AddHealthChecks(): registers the health check framework
➜ AddCheck<>(): adds custom health checks
➜ Endpoint: exposes health status information

✅ Use UseHealthChecks() from the HealthChecks.UI package for a visual dashboard and detailed JSON output.

The AspNetCore.Diagnostics.HealthChecks GitHub repo provides ready-to-use health checks, perfect for monitoring systems like SQL Server, PostgreSQL, Redis, RabbitMQ, and SignalR.

Core Building Blocks of Health Checks:
↳ IHealthCheck
Implement this interface to define custom checks.
↳ HealthCheckOptions
Configure how checks run and how results are reported.
↳ HealthCheckResult
Represents the health state: Healthy, Unhealthy, or Degraded.

Example Use Cases:
1) RemoteHealthCheck: Validates the reachability of external endpoints
2) DatabaseHealthCheck: Confirms database connectivity

3) MemoryHealthCheck: Tracks memory consumption and GC metrics

P.S.

Health checks give you early alerts to potential issues.

They support uptime and help you catch failures before users do.

Using the health check system, build more robust, observable applications.

What have you used health checks for in ASP .NET Core?

———

♻️ Feel free to reshare and empower your network!

🔔 Follow me [Elliot One] and click the bell for more daily insights like this.

#dotnet #csharp