

第67期-基础服务运行健康状况检查

2019年4月20日 09:29

理解状况检查

ASP.NET Core 提供运行状况检查中间件和库，用于报告应用基础结构组件的运行状况。

运行状况检查由应用程序作为 HTTP 终结点公开，以 WEB API 形式公开，可动态配置。

- 检查应用程序的状态，负载均衡器可将流量从失败的实例路由到正常实例，提高可用性。
- 可以监视内存、磁盘和其他物理服务器资源的使用情况来了解是否处于正常状态
- 运行状况检查可以测试应用的依赖项（如数据库和外部服务终结点）以确认是否可用和正常工作。

系统必备

- 引用 Microsoft.AspNetCore.App 元包
- 引用 Microsoft.AspNetCore.Diagnostics.HealthChecks 包。

大多数健康检查实现使用 BeatPulse 开源库，这个库是第三方的，不由微软维护。

AspNetCore.HealthChecks.SqlServer

NUGET包命名：AspNetCore.HealthChecks.*

由微软维护的健康检查包实现

Microsoft.Extensions.Diagnostics.HealthChecks.EntityFrameworkCore

NUGET包命名：Microsoft.Extensions.Diagnostics.HealthChecks.*

基本运行状况检查

对于许多应用，检查处理请求方面的可用性即可发现应用的状态。

服务状态表示：[HealthStatus Enum](#)

```
public class Startup
{
    public void ConfigureServices(IServiceCollection services)
    {
        services.AddHealthChecks();
    }
}
```

```

public void Configure(IApplicationBuilder app, IHostingEnvironment env)
{
    app.UseHealthChecks("/health");

    app.Run(async (context) =>
    {
        await context.Response.WriteAsync("Navigate to /health to  
see the health status.");
    });
}

```

HEALTHCHECK CMD curl --fail <http://localhost:5000/health> || exit

创建自己的健康状态检查服务

实现 IHealthCheck 接口

```

public class ExampleHealthCheck : IHealthCheck
{
    public ExampleHealthCheck()
    {
        // Use dependency injection (DI) to supply any required services to the
        // health check.
    }

    public Task<HealthCheckResult> CheckHealthAsync(
        HealthCheckContext context,
        CancellationToken cancellationToken = default(CancellationToken))
    {
        // Execute health check logic here. This example sets a dummy
        // variable to true.
        var healthCheckResultHealthy = true;

        if (healthCheckResultHealthy)
        {
            return Task.FromResult(
                HealthCheckResult.Healthy("The check indicates a healthy result."));
        }

        return Task.FromResult(
            HealthCheckResult.Unhealthy("The check indicates an unhealthy result."));
    }
}

```

注册运行状况检查服务

```

services.AddHealthChecks().AddCheck<ExampleHealthCheck>("example_health_check");

```

使用 AddCheck 添加到运行状况检查服务。

```
services.AddHealthChecks().AddCheck<ExampleHealthCheck>  
("example_health_check",failureStatus: HealthStatus.Degraded,tags: new[] { "example" });
```

AddCheck 还可以执行 lambda 函数。

```
services.AddHealthChecks().AddCheck("Example", () =>  
HealthCheckResult.Healthy("Example is OK!"), tags: new[] { "example" });
```

使用运行状况检查中间件

```
app.UseHealthChecks("/health");
```

```
app.UseHealthChecks("/health", port: 8000);
```