مفاهیم Job و معرفی Hangfire در دات‌نت

در معماری نرم‌افزار، به‌ویژه در برنامه‌های مبتنی بر وب یا سیستم‌های توزیع‌شده، واژهٔ «Job» به یک واحد پردازشی مستقل اشاره دارد که به‌صورت غیربلاک‌کننده، پس‌زمینه‌ای و خارج از چرخهٔ مستقیم Request/Response اجرا می‌شود. Jobها معمولاً برای انجام عملیات زمان‌بر، وظایف تکرارشونده، پردازش‌های نوبتی یا فعالیت‌هایی که نباید باعث کندی پاسخ‌دهی APIشوند، مورد استفاده قرار می‌گیرند.

یک Job می‌تواند شامل عملیات زیر باشد:

- ارسال پیامک یا ایمیل
- پردازش فایل‌های بزرگ
- هماهنگ‌سازی داده‌ها میان سرویس‌ها
- اجرای زمان‌بندی‌شدهٔ محاسبات دوره‌ای
- اجرای وظایف سنگین بدون تأثیر بر پاسخ‌دهی اپلیکیشن

پردازش‌های پس‌زمینه سبب افزایش کارایی، پایداری سیستم و تجربه کاربری بهتر می‌شوند، زیرا عملیات‌های سنگین از مسیر اصلی اجرای درخواست‌های کاربران جدا می‌شوند.

انواع Job در سیستم‌های پس‌زمینه

1) Fire-and-Forget Job

Jobهایی هستند که بلافاصله پس از فراخوانی، در صف پردازش قرار می‌گیرند و بدون نیاز به انتظار برای نتیجه اجرا می‌شوند. این نوع Job معمولاً برای کارهای سریع و کم‌هزینه استفاده می‌شود.

2) Delayed Job

Jobهایی که پس از گذشت مدت‌زمان مشخصی اجرا می‌شوند.
این مدل برای فعالیت‌هایی مانند ارسال یادآوری، ثبت گزارشات تأخیری و... مناسب است.

3) Recurring Job

Jobهایی که بر اساس الگوی زمان‌بندی مشخص و تکرارشونده اجرا می‌شوند.
برای عملیات‌هایی همچون تهیه نسخه پشتیبان روزانه یا پردازش‌های زمان‌بندی‌شده کاربرد دارند.

4) Continuation Job

Jobهایی که پس از اتمام یک Job دیگر اجرا می‌شوند.
این مدل امکان ساخت زنجیره پردازشی چندمرحله‌ای را فراهم می‌کند.

Hangfire چیست؟

Hangfireیک فریم‌ورک متن‌باز و قدرتمند برای اجرای پردازش‌های پس‌زمینه در دات‌نت است.این فریم‌ورک امکان ایجاد، زمان‌بندی، صف‌بندی، اجرای موازی، مدیریت وضعیت و مانیتورینگJob ها را بدون نیاز به مدیریت دستیThread ها فراهم می‌کند.

ویژگی‌های کلیدی Hangfire

- پشتیبانی از انواع جاب ها Fire-and-Forget ، Delayed، Recurring، Continuation
- ذخیره وضعیتJob ها در Storage دائمی مانندSQL Server ، Redisو...
- داشبورد نظارتی برای مشاهده وضعیت اجرایJob ها
- پشتیبانی از چندین Queue با قابلیت اولویت‌بندی
- مقیاس‌پذیری بالا و مناسب برای سیستم‌هایEnterprise

Hangfireبا استفاده از یک پردازنده پس‌زمینه (Hangfire Server) صف‌ها را پردازش می‌کند وJob ها را به صورت مطمئن، پایدار و قابل رصد اجرا می‌کند.

راه‌اندازی Hangfire در پروژه‌های ASP.NET Core

ثبت سرویس‌ها

برای استفاده از Hangfire در یک پروژه ASP.NET Core ، کافی است سرویس Hangfire و Hangfire Server ثبت شود:

```
var builder = WebApplication.CreateBuilder(args);

builder.Services.AddControllers();

builder.Services.AddHangfire(x =>
    x.UseSqlServerStorage(builder.Configuration.GetConnectionString("Default")));

builder.Services.AddHangfireServer();

var app = builder.Build();

app.UseHangfireDashboard();
app.MapControllers();


app.Run();
```

در این مثال، Hangfire از SQL Server برای ذخیره‌سازی وضعیتJob ها استفاده می‌کند.

ایجادJob ها درHangfire

پس از پیکربندیHangfire ، می‌توان انواعJob ها را ایجاد کرد.

1) Fire-and-Forget

```
BackgroundJob.Enqueue(() => Console.WriteLine("Executed"));
```

2) Delayed Job

```
BackgroundJob.Schedule(() => Console.WriteLine("Executed after 1 minute"),
    TimeSpan.FromMinutes(1));
```

3) Recurring Job

```
RecurringJob.AddOrUpdate("daily-sync",() => service.SyncData(),Cron.Daily);
```

4) Continuation Job

```
BackgroundJob.ContinueWith(jobId,() => Console.WriteLine("Continuation executed"));
```

Queueها و مدیریت اولویت

Hangfireقابلیت تعریف و استفاده ازQueue های مختلف را ارائه می‌دهد.هر Queue می‌تواندJob های مربوط به پردازش‌های خاص یا دارای اولویت متفاوت را در خود جای دهد.

مثال تنظیم Queue درWorker :

```
builder.Services.AddHangfireServer(options =>
{
    options.Queues = new[] { "critical", "default", "low" };
});
```

در این سناریو، Jobهای صف «critical» سریع‌تر و با اولویت بالاتر پردازش خواهند شد.

مدیریت خطاها و Retry

Hangfireبه صورت پیش‌فرض در صورت بروز خطا، Jobرا چندین بار Retry می‌کند.این Retryها در Dashboard قابل مشاهده و مدیریت هستند.

امکان تعیین Retry سفارشی نیز وجود دارد:

```
[AutomaticRetry(Attempts = 3, DelaysInSeconds = new[] { 60 })]
public void Process()
{
  // Processing logic
}
```

بهترین شیوه‌ها در استفاده از Hangfire

برای استفاده بهینه از Hangfire رعایت نکات زیر توصیه می‌شود:

- تفکیکJob ها بر اساس Queue و اولویت
- استفاده از CancellationToken درJob های طولانی
- ثبت Log مناسب درJob ها
- جلوگیری از اجرای عملیات Stateful درJob ها
- تحلیل و پایش Dashboard برای مدیریت Job های Failed
- استفاده از Storage مناسب با حجم پردازش
- پرهیز از اجرای عملیات سنگین با وابستگی بالا به منابع مشترک

جمع‌بندی

Hangfireیک ابزار قدرتمند و پایدار در اکوسیستم دات‌نت برای اجرای پردازش‌های پس‌زمینه است.با پشتیبانی از انواعJob ها، Queueها، Retry، داشبورد نظارتی و امکان ذخیره‌سازی پایدار، گزینه‌ای مناسب برای پروژه‌های Enterprise و سیستم‌های با بار کاری بالا محسوب می‌شود.

درک صحیح مفاهیم Job و استفاده از Hangfire در کنار رعایت الگوهای صحیح طراحی، باعث افزایش پایداری، عملکرد و مقیاس‌پذیری سیستم خواهد شد.

گردآوری و تنظیم: انجمن تخصصی دات نت داتین /آرش مصیر

منبع:

https://docs.hangfire.io/en/latest/