

# 第23期-可复原的EFCore数据库连接

2020年4月22日 9:16

## 数据库连接复原和重试逻辑

在 EF Core 连接中启用重试时，使用 EF Core 执行的每项操作都会成为其自己的可重试操作。如果发生暂时性故障，每个查询和 SaveChanges 的每次调用都会作为一个单元进行重试。

```
public class Startup
{
    public IServiceProvider ConfigureServices(IServiceCollection services)
    {
        services.AddDbContext<CatalogContext>(options =>
        {
            options.UseSqlServer(Configuration["ConnectionString"],
                sqlServerOptionsAction: sqlOptions =>
                {
                    sqlOptions.EnableRetryOnFailure(
                        maxRetryCount: 10,
                        maxRetryDelay: TimeSpan.FromSeconds(30),
                        errorNumbersToAdd: null);
                });
        });
    }
}
```

- maxRetryCount 重试次数,
- maxRetryDelay 两次重试时间差,
- errorNumbersToAdd 那些错误会被重试(Database engine errors)

为 SqlConnection 对象生成连接字符串时参阅 [SqlConnection.ConnectionString](#) 文档：

- ConnectRetryCount： 默认值为 1。 范围为 0 到 255。
- ConnectRetryInterval： 默认值为 10 秒。 范围为 1 到 60。
- Connection Timeout： 默认值为 15 秒。 范围为 0 到 2147483647。

## 自定义执行策略

```
protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
{
    optionsBuilder.UseMyProvider("<connection string>", options =>
        options.ExecutionStrategy(...));
}
```

## 执行策略和事务

InvalidOperationException: 配置的执行策略 "SqlServerRetryingExecutionStrategy" 不支持用户启动的事务。使用由 "DbContext.Database.CreateExecutionStrategy()" 返回的执行策略执行事务（作为一个可回溯单元）中的所有操作。

```
private static void ExecuteWithManualTransaction()
{
    using var db = new BloggingContext();
    var strategy = db.Database.CreateExecutionStrategy();

    strategy.Execute(() =>
    {
        using var context = new BloggingContext();
        using var transaction = context.Database.BeginTransaction();

        context.Blogs.Add(new Blog { Url =
"https://www.xcode.me/dotnet" });
        context.SaveChanges();

        context.Blogs.Add(new Blog { Url =
"https://www.xcode.me/visualstudio" });
        context.SaveChanges();

        transaction.Commit();
    });
}

private static void ExecuteWithManualAmbientTransaction()
{
    using var context1 = new BloggingContext();
    context1.Blogs.Add(new Blog { Url = "https://www.xcode.me/visualstudio" });

    var strategy = context1.Database.CreateExecutionStrategy();

    strategy.Execute(() =>
    {
        using var context2 = new BloggingContext();
        using var transaction = new TransactionScope();

        context2.Blogs.Add(new Blog { Url =
"https://www.xcode.me/dotnet" });
        context2.SaveChanges();

        context1.SaveChanges();

        transaction.Complete();
    });
}
```

```
}
```

## 参考资料

[实现复原 Entity Framework Core SQL 连接](#)

[Entity Framework Core 连接复原官方文档](#)