

第36期-异步查询&工作原理&注释标记

2019年6月18日 10:59

异步查询

当在远程数据库中执行查询时，同步查询可能会阻止当前线程，异步查询可避免阻止线程，这有助于避免冻结客户端界面，异步操作还可以增加 Web 应用程序的吞吐量，可以在数据库操作完成前释放线程去处理其他请求。

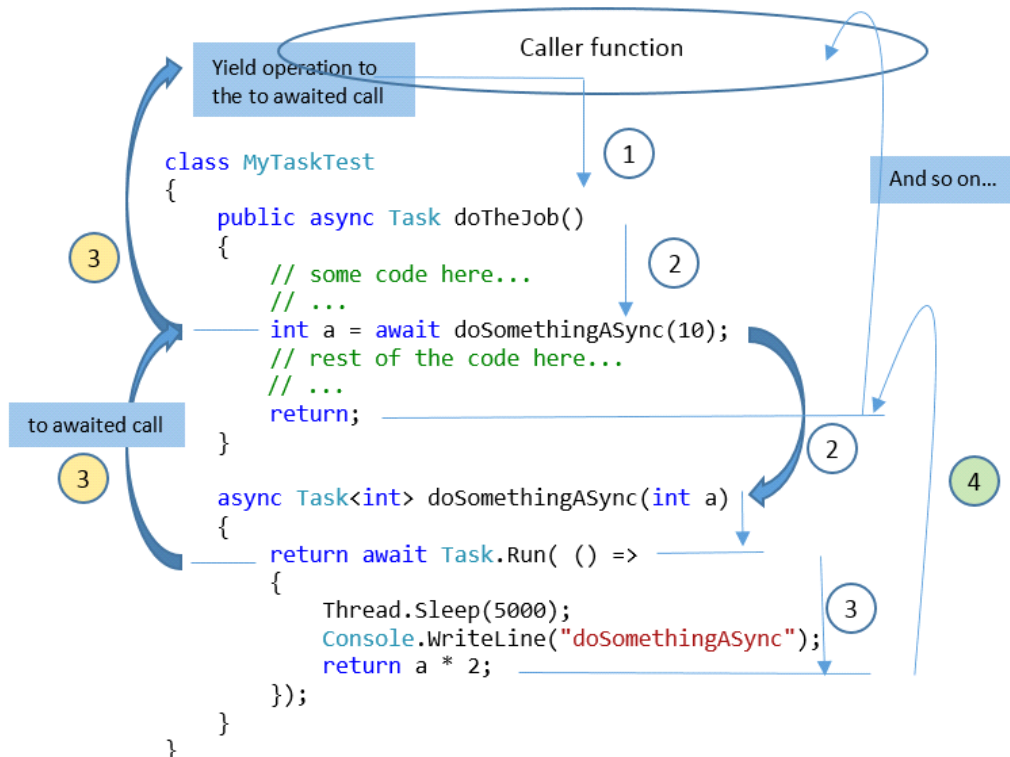
EF Core 不支持在同一上下文实例上运行多个并行操作。应始终等待操作完成，然后再开始下一个操作。这通常是通过在每个异步操作上使用 `await` 关键字完成的。

Entity Framework Core 提供了一组异步扩展方法，可用于执行异步查询。例如：

`ToListAsync()`、`ToArrayAsync()`、`SingleAsync()`，如果看到 `Async` 结尾，一般都是异步方法，这也是微软的命名约定，希望您也遵守。

对于部分 LINQ 运算符（如 `Where(...)`、`OrderBy(...)` 等），没有对应的异步版本，因为这些方法仅用于构建 LINQ 表达式树，而未将查询发送到数据库中执行，只有用到结果时才会去数据库查询，这是 `IQueryable` 天生的特性，之前我们讲过。

```
public async Task<List<Blog>> GetBlogsAsync()
{
    using (var context = new BloggingContext())
    {
        return await context.Blogs.ToListAsync();
    }
}
```



异步编程 <<https://docs.microsoft.com/zh-cn/dotnet/csharp/async>>

查询的工作原理

1. 查询编写的所由表达式由 EF Core 处理，根据不同的数据库提供程序生成与 LINQ 等效的 SQL 语句。
A: 查询到的数据会被缓存，以便每次执行相同数据查询时耗费性能。
2. 查询表达式会传递到数据库提供程序
A: 提供程序会识别出查询的哪些部分可以在数据库中求值进行，哪些只能在客户端进行，哪些根本就没办法进行。
B: 对于能进行数据库查询的操作，生成对应的 SQL 语句。
C: 一个或多个查询会被发送到数据库并返回结果集。这里返回的不是实体对象集合，而是数据集。
3. 对于结果集中的每一项
A: 如果是跟踪查询（默认跟踪），EF 会检查数据项是否已存在于跟踪器中。如果是，则返回现有实体，如果不是，则创建实体，并将实体添加到跟踪器中。
B: 如果这个查询是非跟踪查询，EF 会检查数据是否表示此查询结果集中的现有实体，如果是，则会返回现有实体，如果不是，则会创建新实体并返回该新实体。

调用 LINQ 运算符时，只会构建查询在内存中的表示形式。只有在使用结果时，查询才会发送到数据库。

导致查询发送到数据库的最常见操作如下：

- 在 for 或者 foreach 循环中循环访问结果
- 使用 ToList、ToArray、Single、Count 等操作
- 将查询结果数据绑定到 UI 界面时。

查询注释标记

使用 TagWith() 方法对 LINQ 生成的 SQL 语句进行注释，方便阅读。

```
context.Friends.TagWith("This is my spatial query!")
```