

黄明辉的博客

ASP.NET 异步编程之Async await

本文重点介绍的是.NET Framework4.5 推出的异步编程方案 async await

请先看个5分钟的微软演示的视频：
视频地址：<https://channel9.msdn.com/Blogs/ASP-NET-Site-Videos/async-and-await>

网络上已经有很多文章介绍了这个技术点的应用方式，但是举的例子都是.NET 自带提供的系统异步方法
所以有些同学就看不大懂，如果是非系统自带的，如何实现异步。

实际上，有时候一点就能通，把异步的本质了解明白了，也就懂了。

异步编程的本质就是 **新开任务线程来处理**

这个如何理解呢，请看下文介绍异步编程发展史

```
static void Main() {  
  
    new Thread(Go).Start(); // .NET 1.0开始就有的  
    Task.Factory.StartNew(Go); // .NET 4.0 引入了 TPL  
    Task.Run(new Action(Go)); // .NET 4.5 新增了一个Run的方法  
}
```

这里大家不难看出，.NET 4.5之后引入了 Task.Run方法，实际上呢，异步编程就是通过这个实现的。

了解线程的人也知道，新开一个线程来处理事务，这个很常见，但是在以往，是没办法接收线程里面返回的值的。

所以这时候就该 await 出场了

await，从字面意思，不难理解，就是等待的意思。

执行await的方法必须是async修饰的，并且是Task的类型。异步执行后，返回的信息存储在result属性中。

但并非主进程就会卡在await行的代码上，执行到await方法之后，主线程继续往下执行，无需等待新的线程执行完再继续。

当需要用到新线程返回的result结果时，此时主进程才会等待新线程执行完并返回内容。

也就会说，若无需用到新线程返回的结果，那么主进程不会等待。

async 和await呢，返回类型就3种，void，Task，Task<TResult>

1、void

如果在触发后，你懒得管，请使用 void。

void 返回类型主要用在事件处理程序中，一种称为“fire and forget”（触发并忘记）的活动的方法。除了它之外，我们都应该尽可能是用 Task，作为我们异步方法的返回值。

返回 void，意味着不能 await 该异步方法，即可能出现线程阻塞，并且也无法获取 exception，抛出的异常，通常这些异常会导致我们的程序失败，如果你使用的是 Task 和 Task<Result>，catch 到的异常会包装在属性里面，调用方法就可以从中获取异常信息，并选择正确的处理方式。

2、Task

你如果只是想知道执行的状态，而不需要一个具体的返回结果时，请使用 Task
与void对比呢，是Task可以使用await进行等待新线程执行完毕。而void不需要等待。

公告

昵称：黄明辉
园龄：6年1个月
粉丝：5
关注：1
[+加关注](#)

2020年6月						
日	一	二	三	四	五	六
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4
5	6	7	8	9	10	11

搜索

常用链接

[我的随笔](#)
[我的评论](#)
[我的参与](#)
[最新评论](#)
[我的标签](#)

我的标签

[SignalR\(9\)](#)
[asp.net core\(6\)](#)
[Autofac\(4\)](#)
[MVC\(4\)](#)
[redis\(4\)](#)
[分布式锁\(3\)](#)
[前缀树\(3\)](#)
[SM\(2\)](#)
[国密\(2\)](#)
[ASP.NET\(2\)](#)
[更多](#)

随笔档案

[2020年4月\(1\)](#)
[2020年2月\(3\)](#)
[2019年12月\(3\)](#)
[2019年11月\(1\)](#)
[2019年8月\(1\)](#)
[2019年7月\(1\)](#)
[2019年6月\(7\)](#)
[2019年3月\(2\)](#)

3、Task<TResult>

当你添加 async 关键字后，需要返回一个将用于后续操作的对象，请使用 **Task<TResult>**。

主要有两种方式获取结果值，一个是使用 Result 属性，一个是使用 await。他们的区别在于：如果你使用的是 Result，它带有阻塞性。即在任务完成之前进行访问读取它，当前处于活动状态的线程都会出现阻塞的情形，一直到结果值可用。所以，在绝大多数情况下，除非你有绝对的理由告诉自己，否则都应该使用 await，而不是属性 Result 来读取结果值。

下面我们直接看一些代码就懂了：

```
/// <summary>
/// 添加多条
/// </summary>
/// <param name="list"></param>
/// <returns></returns>
public virtual bool Add(IEnumerable<T> list)
{
    CreateDataBase(); //创建数据库连接
    foreach (T t in list)
    {
        this.Add(t);
    }
    return true;
}

public virtual async Task<bool> AddAsync(IEnumerable<T> list)
{
    return await Task.Run(() => this.Add(list));
}
```

正常情况下，我们约定，异步的方法名均以Async结尾。

以下是服务层调用

```
/// <summary>
/// 日志的json格式字符串 包含 title, url, level,descript
/// </summary>
/// <param name="content"></param>
/// <returns></returns>
public async Task<bool> AddAsync(string content)
{
    using (var mongoDbContext = new Log.DAL.DbContext())
    {
        if (!string.IsNullOrEmpty(content))
        {
            Log.Model.Record model =
Newtonsoft.Json.JsonConvert.DeserializeObject<Log.Model.Record>(content);
            model.AddTime = DateTime.Now;
            var result = await mongoDbContext.Record.AddAsync(model);
            return result;
        }
        else
        {
            return false;
        }
    }
}
```

再接着，进行控制器中的调用

```
public async Task<ActionResult> Add(string content)
{
    var flag = await new Log.Service.RecordService().AddAsync(content); ;
}
```

2018年11月(3)

2018年9月(2)

2018年8月(9)

最新评论

1. Re:ASP.NET Core 在windows服务器安装dotnet-hosting-2.2.6-win 时出现应用程序池停止问题的解决方案
安装后还是不行啊
- 夯
2. Re:ASP.NET 国密加密 SM2-SM4
@SM2把我折磨疯了 你的公钥必须得是SM2的，如果是RSA的，用不了...
- 黄明辉
3. Re:ASP.NET 国密加密 SM2-SM4
@黄明辉 我有公钥，麻烦问一下应该如何用自己的公钥？...
- SM2把我折磨疯了
4. Re:ASP.NET 国密加密 SM2-SM4
@SM2把我折磨疯了 代码里面有生成公钥和私钥的方法，用那个方法生成密钥对...
- 黄明辉
5. Re:ASP.NET 国密加密 SM2-SM4
@刘发源 不需要证书...
- 黄明辉

阅读排行榜

1. ASP.NET 异步编程之Async await(1540)
2. 百度webuploader如何实现秒传与断点续传(1130)
3. ASP.NET SignalR 系列（六）之连接事件(925)
4. ASP.NET SignalR 系列（四）之指定对象推送(889)
5. ASP.NET SignalR 系列（八）之跨域推送(663)

评论排行榜

1. ASP.NET 国密加密 SM2-SM4(6)
2. ASP.NET MVC IOC 之 Autofac 系列开篇(1)
3. ASP.NET Core开发期间部署到IIS自定义主机域名并附加进程调试(1)
4. ASP.NET Core 在windows服务器安装dotnet-hosting-2.2.6-win 时出现应用程序池停止问题的解决方案(1)

推荐排行榜

1. ASP.NET SignalR 系列（四）之指定对象推送(2)
2. ASP.NET SignalR 系列（八）之跨域推送(1)
3. ASP.NET Core+MongoDB（一）(1)
4. ASP.NET SignalR 系列（三）之代码实现(1)

```
        return View();
    }
}
```

下面的代码呢，我们演示一下什么时候需要用到result属性

```
//不需要，因为用了await

public async Task<ActionResult> Detail(int id)
{
    var model = await new DbContext().Record.FirstOrDefaultAsync(m => m.ID == id);
    return View(model);
}

//需要
public async Task<ActionResult> Detail(int id)
{
    var model = new DbContext().Record.FirstOrDefaultAsync(m => m.ID == id).Result;
    return View(model);
}
```

这样，我们异步编程就讲解完了。本文主要是讲解异步是实质，学习这个需要异步有一定的学习和了解。

总结：

- 当你添加 async 关键字后，需要返回一个将用于后续操作的对象，请使用 Task<TResult>;
- 你如果只是想知道执行的状态，而不需要知道具体的返回结果时，请使用 Task;
- 如果在触发后，你懒得管，请使用 void。
- 请尽量优先使用 Task<TResult> 和 Task 作为异步方法的返回类型。
- 用了await，方法必须使用async来修饰。

参考文章：<http://www.cnblogs.com/jesse2013/p/async-and-await.html> 这里介绍了异步的详细发展史。

标签: ASP.NET , 异步编程

好文要顶

关注我

收藏该文

黄明辉

关注 - 1

粉丝 - 5

+加关注

0

0

« 上一篇：[ASP.NET SignalR 系列（九）之源码与总结](#)
» 下一篇：[ASP.NET MVC 允许跨域请求设置](#)

posted @ 2018-09-13 10:29 黄明辉 阅读(1540) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

- 注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#) [网站首页](#)。
- 【推荐】了解你才能更懂你，博客园首发问卷调查，助力社区新升级
- 【推荐】超50万行VC++源码：大型组态工控、电力仿真CAD与GIS源码库
- 【推荐】如何在面试中成长？来看阿里前端终面官的面试心得

最新 IT 新闻:

- 周杰伦在快手发布新歌，能征服 00 后的老铁们吗？
 - 上世纪诞生的游戏卡带，为何没有被 Steam 们取代？
 - 【暗夜明灯】他为何成为“中国最惨创业者”？他是否可以脱身？
 - “最惨创业者”对赌案反思：机械解读条文还是维护机会公平？
 - 618大战：5G推机潮和5G换机潮的冲突
- » 更多新闻...