

C#任务调度框架Quartz.NET

原创 痕迹 CodeShare 1月14日

收录于话题

#C#

10个

什么是定时任务？

最近恰巧有类似的需求, 需要做一个应用程序服务, 每天定时给服务器上传采集的数据。在没有任务框架的使用前提下, 如果我们想要实现类似的需求, 可以自己写一个计时器, 然后24小时运行, 达到指定的时间就运行。但是这样不仅扩展性差, 而且不易于维护。在定时任务这块, 也有非常的多的框架支持, 开箱即用, 那么下面将主要介绍一下Quartz.NET框架。

Github地址: <https://github.com/quartznet/quartznet>

Quartz.NET介绍

Quartz.NET是一个Github开源项目, 用于创建一个免费商业用途的作业调度框架。
注:支持 .NET Core/.NET Standard 2.0 和 .NET 框架 4.6.1 及更晚版本。

Quartz.NET入门

在我们使用定时任务框架的之前, 首先我们需要确定的是, 它大概具备哪些内容。

- 灵活的指定某项任务
- 灵活的启动时间配置
- 传递参数并且监听

了解了这些基础概念之后, 我们便开始学习如何在Quartz.NET当中使用这些功能。

1. 创建一个任务

任务是一个抽象的概念, 实际来说它应该指的是某种任务, 例如: 想要每3秒输入一个字符串, 那么我们就可以这样做。

- 下面演示创建了一个类, 它继承于Quartz中的IJob接口, 实现了Execute方法, 任务就是输出一个字符串。

```
public class TestJob : IJob
{
    public async Task Execute(IJobExecutionContext context)
    {
        await Console.Out.WriteLineAsync($"{DateTime.Now}:Hello!");
    }
}
```



- 创建任务

```
//创建一个工作
IJobDetail job = JobBuilder.Create<TestJob>()
    .WithIdentity("TestJob", "Test")
    .Build();
```



2.配置任务的参数

为任务配置一个触发条件及相关的参数。当该配置的参数一旦达到了条件, 上面的任务将被触发。

- 下面演示创建了一个ITrigger实例, 设置每3秒执行一次

```
//创建一个触发条件
ITrigger trigger = TriggerBuilder.Create()
    .WithIdentity("TestJobTrigger", "Test")
    .WithSimpleSchedule(x =>
    {
        x.WithIntervalInSeconds(3).RepeatForever();
    })
    .Build();
```



3.启动任务

创建一个任务调度器, 将任务和配置参数添加到调度器当中。

```
StdSchedulerFactory factory = new StdSchedulerFactory();
//创建任务调度器
IScheduler scheduler = await factory.GetScheduler();
//启动任务调度器
scheduler.Start();

//将创建的任务和触发器条件添加到创建的任务调度器当中
scheduler.ScheduleJob(job, trigger);
```



4.实际效果

在上面的代码中,我们创建了一个任务输出一段字符串, 然后为该任务创建了一个配置, 每3秒执行一次, 那么实际效果会每间隔3秒输出:

```
2021年1月14日 16:18:22:Hello!
2021年1月14日 16:18:25:Hello!
2021年1月14日 16:18:28:Hello!
2021年1月14日 16:18:31:Hello!
2021年1月14日 16:18:34:Hello!
2021年1月14日 16:18:37:Hello!
2021年1月14日 16:18:40:Hello!
2021年1月14日 16:18:43:Hello!
```



更多配置

- 特定时间触发,重复触发,单次触发, 触发次数,开始时间, 结束时间。这些在框架当中都有该功能
- 触发器表达式 ,例如: 每周一上午, 或每周一、周三、周五等等
下面是一些表达式及其含义的示例

示例 1 - 创建每 5 分钟触发一次的触发器的表达式

```
"0 0/5 * * * ?"
```

示例 2 - 创建每 5 分钟触发一次的触发器的表达式, 在分钟后 10 秒 (即上午 10: 00: 10、上午 10: 05: 10 等) 。

```
"10 0/5 * * * ?"
```

示例 3 - 用于创建触发器的表达式, 该触发器在每个星期三和周五的 10: 30、11: 30、12: 30 和 13: 30 触发。

```
"0 30 10-13 ? * WED,FRI"
```

示例 4 - 创建触发器的表达式, 在每月 5 日到 20 日上午 8 点到上午 10 点之间每半小时触发一次。请注意, 触发器不会于上午 10: 00 触发, 只是在 8: 00、8: 30、9: 00 和 9: 30

```
"0 0/30 8-9 5,20 * ?"
```

构建每天上午 8 点到下午 5 点每隔一分钟触发的触发器:

```
ITrigger trigger = TriggerBuilder.Create()
    .WithIdentity("trigger3", "group1")
    .WithCronSchedule("0 0/2 8-17 * * ?")
    .ForJob("myJob", "group1")
    .Build();
```

构建每天上午 10: 42 触发的触发器:

```
// we use CronScheduleBuilder's static helper methods here
ITrigger trigger = TriggerBuilder.Create()
    .WithIdentity("trigger3", "group1")
    .WithSchedule(CronScheduleBuilder.DailyAtHourAndMinute(10, 42))
    .ForJob(myJobKey)
    .Build();
```



• 监听器: 可以检测你的任务执行状态: 是否工作, 是否错误,是否已完成等
关于具体的使用方法, 在文档中有详细的介绍:
<https://www.quartz-scheduler.net/documentation/quartz-3.x/quick->

start.html

阅读原文 阅读 222

分享

收藏

赞 5

在看 3