



链滴

httpquartz 一个基于 quartz.net 的 http 远程调度系统

作者: [loogn](#)

原文链接: <https://ld246.com/article/1595312205337>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

开发中总是会定时执行一些程序，这个时候任务调度系统总是少不了的，说起任务调度，就不得不说 quartz.NET -----一个.NET 版的任务调度框架。

之前写过一个TaskCaller，不过不是基于Quartz的，没有Quartz的功能强大，这段时间由基于Quartz net重新写了一个httpquartz，文档如下：

[项目地址](#)

项目说明

基于Quartz.Net、Polly的Http远程调度系统，netcore3.1+mysql数据库。

在服务端有HttpJob提供http远程调度，客户端通过对trigger的操作来实现调度任务的管理，通过trigger的JobData传输http请求的相关数据。

功能介绍

- 添加任务
- 编辑任务
- 删除任务
- 暂停任务
- 恢复任务
- 日志查看
- API客户端
- IU管理

服务端部署

1. 新建mysql数据库httpquartz
2. 执行表结构和数据文件/src/HttpQuartz.Server/database/httpquartz.sql
3. 修改appsettings.json和quartz.config中数据库连接信息
4. 配置appsettings.json中SafeClients节点，使信任客户端可以使用HttpAPI
5. 部署HttpQuartz.Server项目(asp.netcore 3.1项目)
6. 用户名是：admin，密码是：123456

API客户端

安装：

`Install-Package HttpQuartz.Client`

使用：

```
//实例化HttpClient，或者通过容器获取，并设置BaseAddress为HttpQuartz.Server的部署地址  
using var client = new HttpClient {BaseAddress = new Uri("http://localhost:5000");};
```

```

var httpQuartzClient = new HttpQuartzClient(client);

//添加新任务(添加触发器), 成功返回success字符串
var result = await httpQuartzClient.ScheduleJob(new TriggerModel()
{
    Key = new TriggerKeyModel("test", "test"),
    StartTime = DateTimeOffset.Now.AddSeconds(30),
    JobData = new JobDataInfo()
    {
        url = "http://www.baidu.com",
    },
    SimpleTrigger = new SimpleTriggerInfo()
    {
        RepeatInterval = TimeSpan.FromSeconds(10)
    },
});

```

UI管理截图

The top screenshot displays the 'HttpQuartz' management interface with a list of jobs:

名称	分组	上下触发时间	开始结束时间	关联数据	操作
keepime.newregister WATING 新注册邮箱提醒	CRON Smart	↑: 2020-07-21 12:00:00 ↓: 2020-07-21 14:00:00	▶: 2020-07-17 18:35:30 ■:	["method":"GET","timeout":"100","url":"http://www.Keepime.com/register","expRefire":"False","expRemove":"False"]	[操作按钮]
mishop.reflashwxtoken WATING 刷新token	SIMPLE Smart	↑: 2020-07-21 13:35:15 ↓: 2020-07-21 13:40:15	▶: 2020-07-17 14:10:15 ■:	["method":"GET","timeout":"100","url":"http://www.mishop.com/refreshwxtoken","expRefire":"False","expRemove":"False"]	[操作按钮]
global.shorturl WATING 全局短网址提醒, 防止过期失效	CRON Smart	↑: 2020-07-21 00:00:00 ↓: 2020-07-22 00:00:00	▶: 2020-07-20 15:22:16 ■:	["method":"GET","timeout":"100","url":"http://www.global-shorturl.com/full7apkey=...","expRefire":"False","expRemove":"False"]	[操作按钮]
test.test PRUSED	CRON Smart	↑: 2020-07-20 17:59:12 ↓: 2020-07-20 18:00:12	▶: 2020-07-16 18:06:24 ■:	["method":"GET","timeout":"100","url":"http://localhost:5000/home/Time","expRefire":"False","expRemove":"False"]	[操作按钮]

The bottom screenshot shows the '添加任务' (Add Task) form with the following JSON configuration:

```

{
  "Key": {
    "Name": "触发器名称 string",
    "Group": "触发器组名 string"
  },
  "Description": "描述 string",
  "Reschedule": "如果key重复是否替换 bool",
  "Priority": "优先级, 默认是5 int?",
  "StartTime": "开始时间 如: 2020-07-07T11:40:00+08:00 DateTimeOffset?",
  "EndTime": "结束时间 DateTimeOffset?",
  "JobData": {
    "method": "请求方法, 默认GET string",
    "url": "请求地址 string",
    "timeout": "请求超时秒数, 默认100秒 int?",
    "expRefire": "异常时是否立即再次触发, 默认false bool",
    "expRemove": "异常时是否移除, 默认false bool"
  },
  "SimpleTrigger": {
    "RepeatInterval": "重复时间间隔 TimeSpan",
    "RepeatCount": "重复次数, 默认-1是永久重复 int",
    "MisfireInstruction": "失火指令 int
-1: IgnoreMisfirePolicy忽略其他指令, 有资源就发出所有MisFire任务
1: FireNow: 忽略已经MisFire的任务, 并立即执行调度 (只执行一次调度的默认值)
4个Trigger最多可以存在1个. CRON在线生成工具

```

相关

[Quartz.Net项目](#)

[Polly项目](#)

[LayUI项目](#)