

TÀI LIỆU HƯỚNG DẪN HANGFIRE TRONG ASP.NET

1. Cài đặt và cấu hình Hangfire:

Bước 1: Cài đặt gói package Hangfire:

Bước 2: Trong SQL Server, tạo mới database “Lab.Hangfire”

Bước 3: Khai báo chuỗi kết nối đến database

Bước 4: Đăng ký service Hangfire trong Startup.cs

```
public class Startup
{
    0 references
    public Startup(IConfiguration configuration)
    {
        Configuration = configuration;
    }

    2 references
    public IConfiguration Configuration { get; }

    // This method gets called by the runtime. Use this method to add services to the container.
    0 references
    public void ConfigureServices(IServiceCollection services)
    {
        services.AddHangfire(x => x.UseSqlServerStorage(Configuration.GetConnectionString("DefaultConnection")));
        services.AddHangfireServer();

        services.AddControllers();
    }
}
```

```
[Obsolete]
0 references
public void Configure(IApplicationBuilder app, IBackgroundJobClient backgroundJobs, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }

    app.UseHttpsRedirection();

    app.UseRouting();

    app.UseAuthorization();

    // use Hangfire Dashboard
    app.UseHangfireDashboard();

    app.UseEndpoints(endpoints =>
    {
        endpoints.MapControllers();
    });
}
```

Bước 5: Cấu hình ứng dụng không khởi động browser khi run project (nâng cao)

```
launchSettings.json* X
Schema: http://json.schemastore.org/launchsettings.json
1  {
2    "$schema": "http://json.schemastore.org/launchsettings.json",
3    "iisSettings": {
4      "windowsAuthentication": false,
5      "anonymousAuthentication": true,
6      "iisExpress": {
7        "applicationUrl": "http://localhost:61991",
8        "sslPort": 44393
9      }
10   },
11   "profiles": {
12     "IIS Express": {
13       "commandName": "IISExpress",
14       "launchBrowser": false,
15       "launchUrl": "hangfire",
16       "environmentVariables": {
17         "ASPNETCORE_ENVIRONMENT": "Development"
18       }
19     },
20     "lab.hangfire.net": {
21       "commandName": "Project",
22       "launchBrowser": false,
23       "launchUrl": "hangfire",
24       "applicationUrl": "https://localhost:5001;http://localhost:5000",
25       "environmentVariables": {
26         "ASPNETCORE_ENVIRONMENT": "Development"
27       }
28     }
29   }
30 }
```

Bước 6: Cấu hình ứng dụng chạy background khi đã tắt cửa sổ console (nâng cao)

```
lab.hangfire.net.csproj* X
<Project Sdk="Microsoft.NET.Sdk.Web">
  <PropertyGroup>
    <TargetFramework>netcoreapp3.1</TargetFramework>
    <OutputType>WinExe</OutputType>
    <DisableWinExeOutputInference>True</DisableWinExeOutputInference>
  </PropertyGroup>
  <ItemGroup>
    <PackageReference Include="Hangfire" Version="1.7.32" />
    <PackageReference Include="MailKit" Version="2.8.0" />
    <PackageReference Include="MimeKit" Version="2.9.1" />
  </ItemGroup>
</Project>
```

2. Khởi tạo background job và sử dụng:

- Có 3 loại job trong hangfire:
 - a. Thực thi ngay lập tức

```
BackgroundJob.Enqueue(() => Console.WriteLine($"Hello world"))
```

b. Thực thi sau một khoảng trễ thời gian quy định:

```
BackgroundJob.Schedule(() => Console.WriteLine($"Hello world"),  
TimeSpan.FromSeconds(10))
```

c. Thực thi theo lịch trình đặt trước

```
var manager = new RecurringJobManager();  
manager.AddOrUpdate("job1", () => Console.WriteLine($"Hello world"), "30 12 15 * *");
```

d. Thực thi tuần tự

```
var jobId = BackgroundJob.Enqueue(() => Goodbye(userName));  
BackgroundJob.ContinueJobWith(jobId, () => Console.WriteLine($"Hello world"),);
```

3. Cấu hình CRON:

- Trong CRON có 5 tham số chính, được thể hiện bằng các dấu (*):

Công thức:

***[minute:0-59] [hour:0-23] [day:1-31] [month:1-12] [dayOfWeek:0-7]**

```
var manager = new RecurringJobManager();  
manager.AddOrUpdate("job1", () => Console.WriteLine($"Hello world"), "30 12 15 * *");
```

Giải thích: Cứ mỗi tháng, vào ngày 15 lúc 12 giờ 30 phút job trên sẽ chạy.

a. Các trường hợp với tham số minute (tham số thứ 1):

- TH1: Quy định tại phút thứ n của mỗi giờ/ngày/tháng thì job sẽ chạy.

```
var manager = new RecurringJobManager();  
manager.AddOrUpdate("job1", () => Console.WriteLine($"Hello world"), "30 * * * *");
```

Giải thích: Cứ mỗi ngày vào phút thứ 30 của mỗi giờ job trên sẽ chạy.

- TH2: Quy định sau n phút job sẽ thực thi

```
var manager = new RecurringJobManager();  
manager.AddOrUpdate("job1", () => Console.WriteLine($"Hello world"), "%/5 * * * *");
```

Giải thích: Cứ mỗi 5 phút job sẽ thực thi một lần.

- TH3: Quy định sau mỗi phút job sẽ thực thi

```
var manager = new RecurringJobManager();  
manager.AddOrUpdate("job1", () => Console.WriteLine($"Hello world"), "* * * * *");
```

Giải thích: Cứ sau một phút, job sẽ thực thi.

b. Các trường hợp đối với tham số hour (tham số thứ 2):

- TH1: Quy định sau mỗi giờ job sẽ thực thi:

```
var manager = new RecurringJobManager();  
manager.AddOrUpdate("job1", () => Console.WriteLine($"Hello world"), "* * * * *");
```

Giải thích: Cứ sau mỗi một giờ, job sẽ thực thi.

- TH2: Quy định vào giờ thứ n job sẽ thực thi

```
var manager = new RecurringJobManager();  
manager.AddOrUpdate("job1", () => Console.WriteLine($"Hello world"), "* * 12 * * *");
```

Giải thích: Cứ đến 12h mỗi ngày, job sẽ thực thi.

- TH3: Quy định cứ sau n giờ job sẽ thực thi

```
var manager = new RecurringJobManager();  
manager.AddOrUpdate("job1", () => Console.WriteLine($"Hello world"), "* * */12 * * *");
```

Giải thích: Cứ sau 12 tiếng, job sẽ thực thi 1 lần

c. Các trường hợp đối với tham số day (tham số thứ 3):

- TH1: Quy định sau mỗi ngày job sẽ thực thi:

```
var manager = new RecurringJobManager();  
manager.AddOrUpdate("job1", () => Console.WriteLine($"Hello world"), "* * * * *");
```

Giải thích: Mỗi ngày job sẽ thực thi 1 lần

- TH2: Quy định vào ngày thứ n job sẽ thực thi:

```
var manager = new RecurringJobManager();  
manager.AddOrUpdate("job1", () => Console.WriteLine($"Hello world"), "* * 15 * *");
```

Giải thích: Vào ngày 15 mỗi tháng, job sẽ thực thi.

- TH3: Quy định sau n ngày job sẽ thực thi:

```
var manager = new RecurringJobManager();  
manager.AddOrUpdate("job1", () => Console.WriteLine($"Hello world"), "* * */5 * *");
```

Giải thích: Sau 5 ngày, job thực thi 1 lần.

- TH4: Quy định job thực thi từ ngày a-b

```
var manager = new RecurringJobManager();  
manager.AddOrUpdate("job1", () => Console.WriteLine($"Hello world"), "* * 1-15 * *");
```

Giải thích: Từ ngày 1-15 trong mỗi tháng, job sẽ thực thi mỗi ngày 1 lần.

- TH5: Quy định job chạy đầu tháng

```
var manager = new RecurringJobManager();
manager.AddOrUpdate("job1", () => Console.WriteLine($"Hello world"), "** * 1 * *");
```

Giải thích: Vào ngày 1 của mỗi tháng, job sẽ chạy.

- TH6: Quy định job chạy cuối tháng

```
var manager = new RecurringJobManager();
manager.AddOrUpdate("job1", () => Console.WriteLine($"Hello world"), "** * L * *");
```

Giải thích: Vào ngày cuối cùng của mỗi tháng, job sẽ chạy.

d. Các trường hợp đối với tham số month (tham số thứ 4):

- TH1: Quy định job chạy mỗi tháng

```
var manager = new RecurringJobManager();
manager.AddOrUpdate("job1", () => Console.WriteLine($"Hello world"), "* * * * *");
```

Giải thích: Cứ mỗi tháng, job sẽ chạy 1 lần.

- TH2: Quy định tháng n job sẽ chạy

```
var manager = new RecurringJobManager();
manager.AddOrUpdate("job1", () => Console.WriteLine($"Hello world"), "* * * 5 *");
```

Giải thích: Job sẽ chạy vào, mỗi phút, mỗi giờ, mỗi ngày trong tháng 5

- TH3: Quy định sau n tháng job sẽ chạy

```
var manager = new RecurringJobManager();
manager.AddOrUpdate("job1", () => Console.WriteLine($"Hello world"), "* * * */3 *");
```

Giải thích: Sau 3 tháng, job sẽ chạy 1 lần

- TH4: Quy định job chạy từ tháng a-b

```
var manager = new RecurringJobManager();  
manager.AddOrUpdate("job1", () => Console.WriteLine($"Hello world"), "** * * 1-3 *");
```

Giải thích: Từ tháng 1-3, job sẽ thực thi mỗi ngày, mỗi giờ, mỗi phút

e. Các trường hợp đối với tham số dayOfWeek (tham số thứ 5):

- TH1: Quy định job chạy mỗi ngày trong tuần

```
var manager = new RecurringJobManager();  
manager.AddOrUpdate("job1", () => Console.WriteLine($"Hello world"), "* * * * *");
```

Giải thích: Job sẽ chạy mỗi ngày trong tuần, và chạy trong mỗi giờ, mỗi phút của ngày đó

- TH2: Quy định job chạy vào thứ n

```
var manager = new RecurringJobManager();  
manager.AddOrUpdate("job1", () => Console.WriteLine($"Hello world"), "* * * * 5");
```

Giải thích: Job sẽ chạy vào thứ 6 của mỗi tháng, và chạy trong mỗi giờ, mỗi phút của ngày đó

- TH3: Quy định job chạy vào thứ n của tuần thứ m trong tháng

```
var manager = new RecurringJobManager();  
manager.AddOrUpdate("job1", () => Console.WriteLine($"Hello world"), "* * * * 5");
```

Giải thích: Job sẽ chạy vào thứ 6 của tuần thứ 2 trong tháng, và chạy trong mỗi giờ, mỗi phút của ngày đó.

TÀI LIỆU THAM KHẢO:

[1]. Hangfire - <https://www.hangfire.io/>

[2]. Cron expressions - https://en.wikipedia.org/wiki/Cron#CRON_expression