

USING BACKGROUND PROCESSING TO BUILD SCALABLE APPLICATIONS WITH HANGFIRE

Lab 9

Goal

As our application continues to grow and we create more and more jobs, the logging framework we put in place may not be sufficient. In this lab we will take a deeper dive into logging and update our logging to make it easier to find what we are looking for (if we were to ever have an error that we would need to dig up more information on).

In the first part of the lab we will be updating our logging with a MappedDiagnostics and GlobalDiagnostics Context so that we can easily find all the logs for each instance of a job.

1. Open the BaseLemmingJob in Core.Jobs and add the following protected variable:

```
protected readonly Guid Id = Guid.NewGuid();
```

2. Add a constructor to the BaseLemmingJob

```
protected BaseLemmingJob()  
{  
    MappedDiagnosticsContext.Set("JobId", Id.ToString("N"));  
}
```

3. Next open the Bootstrap class in Core.Configuration and add the following to the beginning of the BootstrapNLog method:

```
GlobalDiagnosticsContext.Set("ServiceInstanceId", Guid.NewGuid().ToString("N"));
```

4. Run your project by hitting F5 and queue some jobs at random.
5. Open SQL Server Management Studio or SQL Server Object Explorer (View -> SQL Server Object Explorer) in Visual Studio and look at the EventLog table with the following query:

```
USE LemmingSchedulingSystem
```

```
SELECT TOP 100 *
```

```
FROM dbo.EventLog
```

```
ORDER BY Timestamp DESC
```

6. Notice that can now identify all messages that were logged for a particular instance of a job using the JobId column and the server that the job ran on with the ServiceInstanceId column

Results		Messages								
		Id	Logger	Timestamp	Level	Message	Exception	StackTrace	ServiceInstanceId	JobId
1	31		Core.Jobs.BaseLemmingJob	2016-03-24 23:21:10.957	Info	Unclogging some toilets...		b2773f9e00f349328022147930bfa4d3	b07831edb01c450bb0cb10ee6755021c	
2	30		Core.Jobs.BaseLemmingJob	2016-03-24 23:21:11.700	Info	Running job of type PlumbingJob.		h2773f9e00f349328022147930bfa4d3	h07831adb811c458b0c1b16ee8f75821c	
3	29		Core.Jobs.BaseLemmingJob	2016-03-24 23:21:11.650	Info	Running job of type WeldingJob.		b2773f9e00f349328022147930bfa4d3	1/a0b16c65b8fa78af4d9e30b7e27ef4	
4	28		Core.Jobs.BaseLemmingJob	2016-03-24 23:21:07.310	Info	Building a house...		b2773f9e00f349328022147930bfa4d3	41fd795cc49a47c1a0a13a9134249ab	
5	27		Core.Jobs.BaseLemmingJob	2016-03-24 23:21:05.833	Info	Running job of type SayHelloJob		b2773f9e00f349328022147930bfa4d3	6daf1371c8d644c78889c357c2011c2	
6	26		Core.Jobs.BaseLemmingJob	2016-03-24 23:21:05.610	Info	Painting some walls...		b2773f9e00f349328022147930bfa4d3	80e63ead2a4fd4991ebdc3aff9363	
7	25		Core.Jobs.BaseLemmingJob	2016-03-24 23:20:56.173	Info	Running job of type PaintingJob.		b2773f9e00f349328022147930bfa4d3	80e63ead2a4fd4991ebdc3aff9363	
8	24		Core.Jobs.BaseLemmingJob	2016-03-24 23:20:10.273	Info	Building a house...		b2773f9e00f349328022147930bfa4d3	1496c45254b547cb0f1192eeb301205	
9	23		Core.Jobs.BaseLemmingJob	2016-03-24 23:19:28.607	Info	Building a house		h2773f9e00f349328022147930bfa4d3	7e5cb3aef8a4677b4d5935c7efa37524c	
10	22		Hangfire.SqlServer.SqlServerStorage	2016-03-24 23:17:30.823	Info	Hangfire SQL objects installed.		3c0132e0b1854936a48da80dff65371		
11	21		Hangfire.SqlServer.SqlServerStorage	2016-03-24 23:17:30.723	Info	Start installing Hangfire SQL objects...		3c0132e0b1854936a48da80dff65371		

Next we will log some messages to Slack. We have already setup a slack server for you to use and setup the channel. If you want more information on setting up a WebHook in Slack please review:

<https://api.slack.com/incoming-webhooks>

7. Add a new folder to Core called Model
8. Add a new class to Core.Model call SlackPostModel

```
public class SlackPostModel
{
    [JsonProperty("text")]
    public string Text { get; set; }
}
```

9. Add the following using statement to this class:

```
using Newtonsoft.Json;
```

10. Add a new folder to Core called Extensions and add a new class to this called SlackTarget.

```
[Target("Slack")]
public sealed class SlackTarget : TargetWithLayout
{
    [RequiredParameter]
    public string MessagePrefix { get; set; }

    [RequiredParameter]
    public string ChannelUrl { get; set; }

    protected override void Write(LogEventInfo logEvent)
    {
        var logMessage = Layout.Render(logEvent);

        if (!string.IsNullOrEmpty(MessagePrefix))
        {
            logMessage = $"{MessagePrefix} - {logMessage}";
        }

        var json = JsonConvert.SerializeObject(new SlackPostModel() { Text =
logMessage });

        var httpClient = new HttpClient();

        httpClient.DefaultRequestHeaders.ExpectContinue = false;
        httpClient.DefaultRequestHeaders.ConnectionClose = true;

        var stringContent = new StringContent(json);
        var response = httpClient.PostAsync(ChannelUrl, stringContent).Result;

        try
        {
            response.EnsureSuccessStatusCode();
        }
        catch (Exception)
        {
            throw;
        }
    }
}
```

11. Add the following using statements to this class:

```
using System.Net.Http;
using Core.Model;
using Newtonsoft.Json;
using NLog;
using NLog.Config;
using NLog.Targets;
```

12. Open the Bootstrap class in Core.Configuration and add the following lines to the BootstrapNLogMethod prior to last line (`LogManager.Configuration = config;`)

```
var slackTarget = new SlackTarget() {ChannelUrl =
"https://hooks.slack.com/services/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",
MessagePrefix = Environment.MachineName};

config.AddTarget("slack", slackTarget);
config.LoggingRules.Add(new LoggingRule("...", LogLevel.Warn, slackTarget));
```

We have now successfully configured NLog to log Warnings to Slack!

13. Open up your favorite job and use the `Logger.Warn` to add a custom warning message. I have chosen the `BuildHouseWithFriendWithName` job.

Example: `Logger.Warn($"I'm building a house with {friendName}. This may not really be my friend.");`

14. Run your project by pressing F5 and queue the job that you updated with a Warning message and take note of the Slack channel on the projector!

Today



incoming-webhook BOT 11:51 PM

WIN8-VIRTUAL - 2016-03-24 23:51:00.1357|WARN|Core.Jobs.BaseLemmingJob|I'm building a house with David. This may not really be my friend.

15. P.S. We have also configured a `ColoredConsoleTarget` for you. Take a look at the messages in the console of your Hangfire server application!

```
2016-03-24 23:50:44.8872|DEBUG|Hangfire.Server.Worker|Background process 'Worker
#ae98af6f' started.
2016-03-24 23:50:44.9249|DEBUG|Hangfire.Server.DelayedJobScheduler|Background pr
ocess 'DelayedJobScheduler' started.
2016-03-24 23:50:45.0476|DEBUG|Hangfire.SqlServer.ExpirationManager|Removing out
dated records from table 'Set'...
2016-03-24 23:50:44.9874|DEBUG|Hangfire.SqlServer.ExpirationManager|Removing out
dated records from table 'Set'...
2016-03-24 23:50:45.3031|DEBUG|Hangfire.SqlServer.ExpirationManager|Removing out
dated records from table 'Hash'...
2016-03-24 23:50:45.3446|DEBUG|Hangfire.SqlServer.ExpirationManager|Removing out
dated records from table 'Hash'...
2016-03-24 23:51:00.1357|WARN|Core.Jobs.BaseLemmingJob|I'm building a house with
David. This may not really be my friend.
2016-03-24 23:51:00.2036|INFO|Core.Jobs.BaseLemmingJob|Running job of type Paint
ingJob.
2016-03-24 23:51:00.3195|INFO|Core.Jobs.BaseLemmingJob|Running job of type Plumb
ingJob.
2016-03-24 23:51:00.9049|INFO|Core.Jobs.BaseLemmingJob|Building a house with my
friend David!
2016-03-24 23:51:05.8417|INFO|Core.Jobs.BaseLemmingJob|Painting some walls...
2016-03-24 23:51:09.0914|INFO|Core.Jobs.BaseLemmingJob|Unclogging some toilets...
2016-03-24 23:51:45.6854|INFO|Core.Jobs.BaseLemmingJob|Building a house...
```

This completes Lab 09.