# Easy Mode Guide LoggerElasticSearch

## Download the Example Project

<https://github.com/Davidfcr/LoggerWorkshop>

## Connect Serilog to ElasticSearch

First, install the next NuGets:

1. Serilog.AspNetCore
2. Serilog.Settings.Configuration
3. Serilog.Sinks.ElasticSearch

Second, head to program.cs and updated it with this lines to setup the logger.

public class Program

{

    public static IConfiguration Configuration { get; } = new ConfigurationBuilder()

        .SetBasePath(Directory.GetCurrentDirectory())

        .AddJsonFile("appsettings.json", optional: false, reloadOnChange: true)

        .Build();

    public static void Main(string[] args)

    {

        Log.Logger = new LoggerConfiguration()

            .ReadFrom.Configuration(Configuration)

            .CreateLogger();

        try

        {

            Log.Information("Getting the motors running...");

            BuildWebHost(args).Run();

        }

        catch (Exception ex)

        {

            Log.Fatal(ex, "Host terminated unexpectedly");

        }

        finally

        {

            Log.CloseAndFlush();

        }

    }

    public static IWebHost BuildWebHost(string[] args) =>

        WebHost.CreateDefaultBuilder(args)

               .UseStartup<Startup>()

               .UseConfiguration(Configuration)

               .UseSerilog()

               .Build();

}

This code add the configuration to use the environment setup up on the appsettings.json, and the 3 lines at the start of the Main method tells the application to activate Serilog as the logger. In the BuildWebHost method we need to add the Serilog service as well.

Next up, add a new appsettings.json file and configure Serilog using these parameters, if you created the table manually, wipe out the autoCreateSqlTable property:

{

  "Serilog": {

    "MinimumLevel": "Information",

"Enrich": ["FromLogContext"],

    "WriteTo": [

      {

        "Name": "Elasticsearch",

        "Args": {

          "nodeUris": "http://localhost:9200",

"autoRegisterTemplate": "true"

        }

      }

    ]

  },

}

All ready. Start logging using the next line wherever you want to send messages into the Database.

Log.Information("Pong received");

You can use the logging structured objects like {@object} in the message template.

Log.Information("Total of {@div} Pong sended ", div);

Log.Information("Pong couldn’t be sended back, Error: {@div}", div);

## Execute the Server

1. Initialize the Docker Service
2. Open Docker Application
3. Download and use the next container:

<https://hub.docker.com/r/arcseldon/elasticsearch-kibana-marvel-sense/>

1. Run the docker container:

## Create the Index Pattern

Connect to <http://localhost:5601> and head to Settings menu options and then Indices tab. Once you are in the menu **Configure an index pattern**, create a pattern using the default name **logtash-\***

## Enjoy

In the Discover menu option, the index pattern is showing the latest records. Now we have the information in Elastic and can start working around with the Visualize and Dashboard menu options.