

# .NET Logging Done Right: An Opinionated Approach Using Serilog

---

GOALS OF GOOD LOGGING / INTRODUCING SERILOG



**Erik Dahl**

PRINCIPAL ARCHITECT

@dahlsailrunner [www.knowyourtoolset.com](http://www.knowyourtoolset.com)



# Overview



Define goals of good logging

What should our end state look like?

Define objectives for logging framework

Introduce Serilog and key concepts of it

- Not a Serilog deep dive
- Jason Roberts: Modern Structured Logging with Serilog and Seq

Course outline



# Goals of Logging

## Improve “Fixability”

Full details in the logs

Easily accessible

Minimize “Can’t reproduce”

## Understand Applications

Most-used features

Speed of key activities

Time(s) for max concurrency

Frequency of errors

Key customer usage

## Prioritize Work

What to fix first

What to speed up

Deflect squeaky wheels

Base decisions on usage and/or customer



# Searching, Exploring, and Viewing Details

**Flog Viewer** Error Logs Performance Logs About

Search Criteria HIDE CRITERIA

Machine Name	Layer	User	Begin	Like
ALL	ALL	ALL	8/4/2017 12:00 AM	
DESKTOP-AEDILCJ	Angular/Client API ClientJS MVC WebForms WpfClient	BobSmith@email.com edahl joe@nowhere.com	End	Not Like

Show at most 100 entries

**HIT IT!**

18 08-04-17 07:21:43 Todos - WpfClient TodoApplication.MainWindow->btnAlert\_Click edahl DESKTOP-AEDILCJ

Message: Something really bad happened!!

Exception

Exception: System.Exception  
Message: Something really bad happened!!  
HResult: -2146233088  
Source: TodoApplication  
StackTrace: at TodoApplication.MainWindow.btnAlert\_Click(Object sender, RoutedEventArgs e) in C:\Users\edahl\Downloads\TodoApplication\TodoApplication\MainWindow.xaml.cs:line 30  
at System.Windows.RoutedEventHandlerInfo.InvokeHandler(Object target, RoutedEventArgs routedEventArgs)  
at System.Windows.EventRoute.InvokeHandlersImpl(Object source, RoutedEventArgs args, Boolean reRaised)

**SQL Server**  
Custom viewer

**kibana** Discover Visualize Dashboard Settings

error\* 24 hits

August 5th 2017, 00:00:00.000 - August 5th 2017, 23:59:59.999 — by 30 minutes

Count

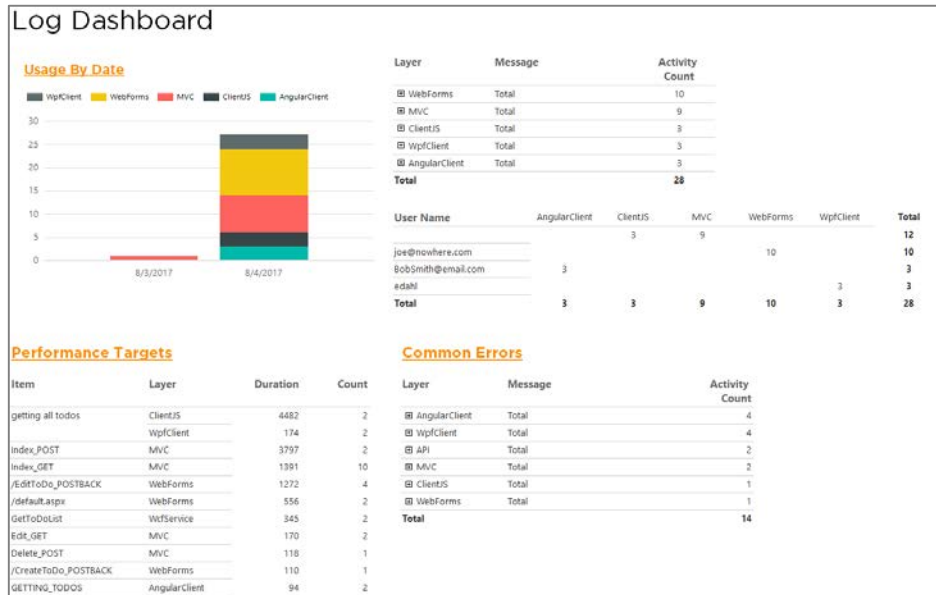
@timestamp

Time	Layer	Location	Hostname	Username	Message
August 5th 2017, 09:12:49.091	ClientJS	https://samplemvc.knowyourtoolset.com/jstodos			failed updating todo
August 5th 2017, 09:12:49.055	API	PUT api/Todos/{id}	DESKTOP-AEDILCJ		String or binary data was truncated. The statement has been terminated.
August 5th 2017, 09:04:31.450	ClientJS	https://samplemvc.knowyourtoolset.com/jstodos			failed updating todo
August 5th 2017, 09:04:31.423	API	PUT api/Todos/{id}	DESKTOP-AEDILCJ		String or binary data was truncated.

**Elasticsearch**  
Kibana: Discovery is in the box!

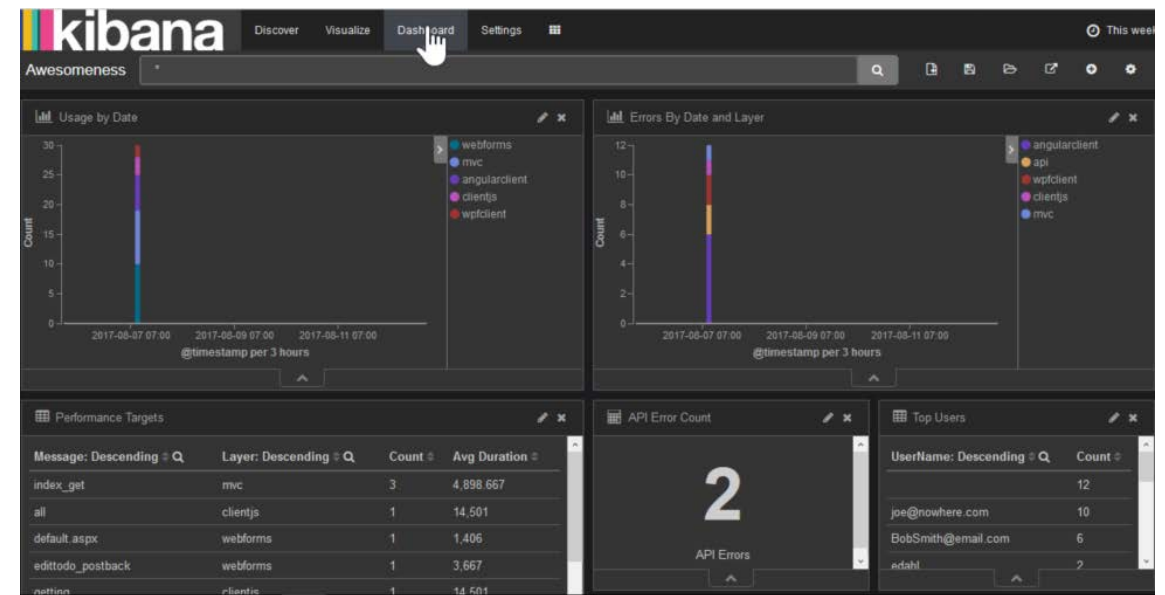


# Dashboards for Deeper Insights



## SQL Server

Custom Reporting Services report



## Elasticsearch

Kibana: Visualizations and Dashboards are in the box!



# Objectives

Consistently good information

Minimize impact to application code

Shield exception details in client / server communications

Enable developers to focus on application code and business logic, not logging

Flexibility in where we write our log entries

Support log writing from any application



# Serilog

(<https://serilog.net>)

## Structured logging framework

Provides us all the flexibility we're after

“Sinks” are places to write log entries – and there are LOTS of options here

### We will leverage:

- Structured nature of framework
- Sinks

### We will not use:

- Levels
- Enrichers



Create Framework

Apply Framework

Choose Sink





Create Framework

**Core logger using Serilog**  
**Write to file sinks to keep it simple**  
**Give some love to data access – want params in logs!**

Apply Framework

Choose Sink



Create Framework

**Core logger using Serilog**  
**Write to file sinks to keep it simple**  
**Give some love to data access – want params in logs!**

Apply Framework

**ASP.Net WebForms & MVC (no JavaScript)**  
**WebAPI**  
**JavaScript (simple) and Angular 4**  
**WPF and WCF**  
**ASP.Net Core**

**Introduce app and code**  
**Apply logging**  
**Re-run app and review logs**

Choose Sink



Create Framework

**Core logger using Serilog**  
**Write to file sinks to keep it simple**  
**Give some love to data access – want params in logs!**

Apply Framework

**ASP.Net WebForms & MVC (no JavaScript)**  
**WebAPI**  
**JavaScript (simple) and Angular 4**  
**WPF and WCF**  
**ASP.Net Core**

**Introduce app and code**  
**Apply logging**  
**Re-run app and review logs**

Choose Sink

**Evaluate options**  
**Write to SQL Server and Elasticsearch**  
**Perform review and analysis of log entries**  
**Visualize data**

