Logging

Übung 9

Serilog

- https://github.com/serilog/serilog/wiki/Structured-Data
- Wenn wir ein Object Foo serialisieren wollen:

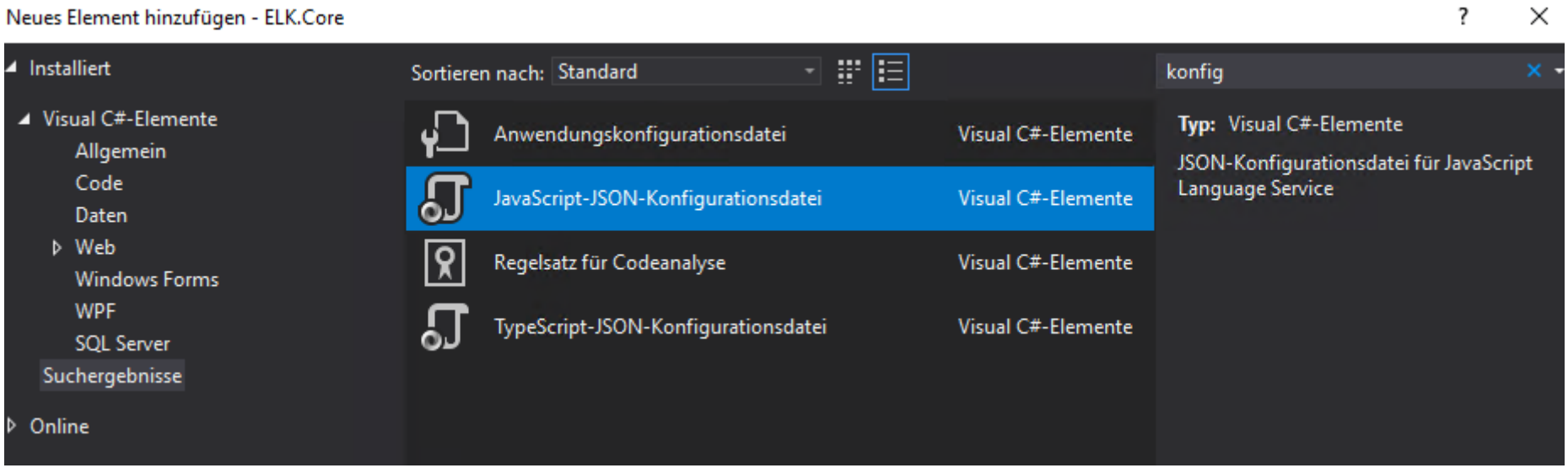
```
Log.Error("{@Foo}", foo);
```

Setup

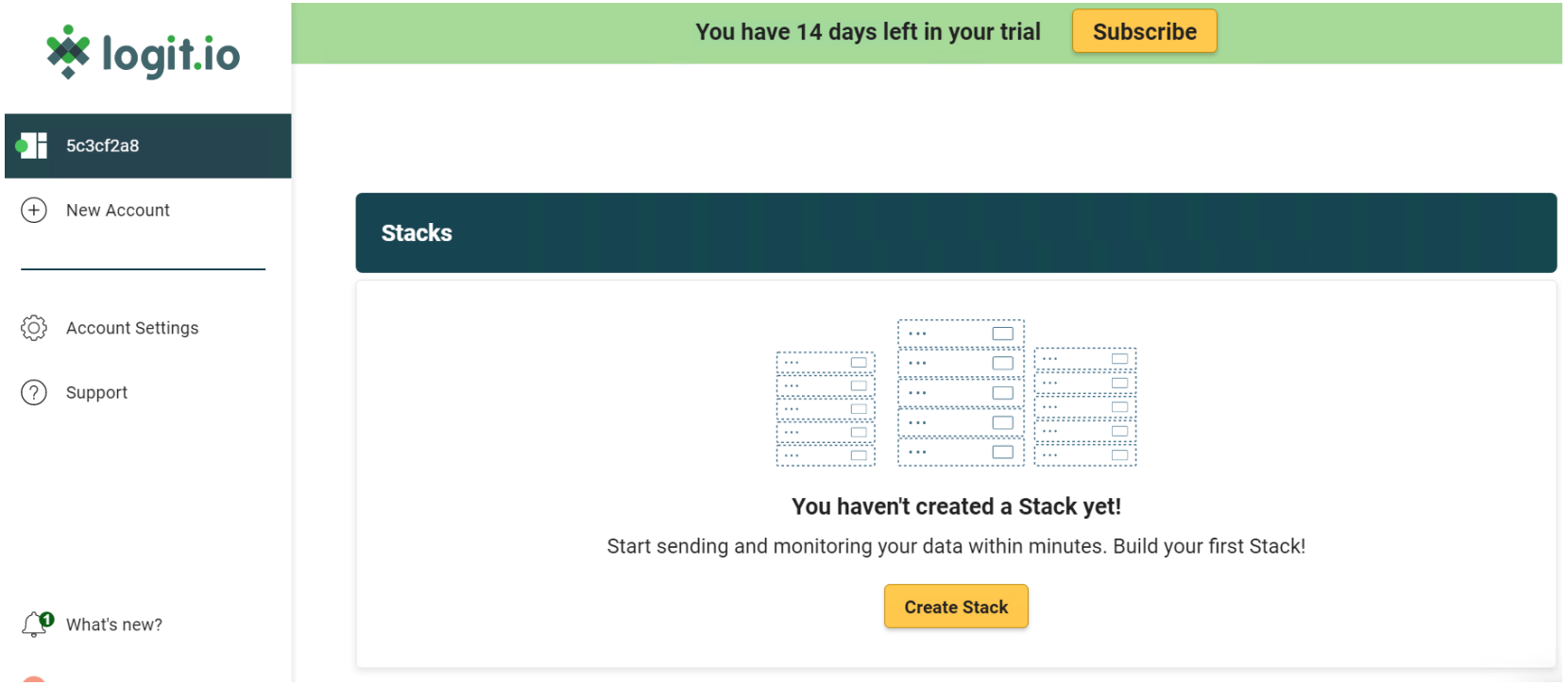
- 14 Tage Test-Account anlegen: logit.io
- Projekt ELK.Core
- Folgende NuGet Pakete sind relveant:
  - Microsoft.Extensions.Configuration.Json
  - Serilog
  - Serilog.Settings.Configuration
  - Serilog.Sinks.Network

appsetting.json hinzufügen

- Über Projekt > Hinzufügen > Neues Element



Neuen Stack anlegen



Konfiguration extrahieren

- https://logit.io/sources/configure/dotnetcore
- Konfiguration kopieren
- Mit „Launch Kibana“ im Cockpit kommt man zu den Logs

FH University of Applied Sciences TECHNIKUM WIEN

Setup

```
var configuration = new ConfigurationBuilder()
    .AddJsonFile("appsettings.json")
    .Build();

Log.Logger = new LoggerConfiguration()
    .ReadFrom.Configuration(configuration)
    .CreateLogger();
```

Aufgabe 1

- Erstellen von 2 Logg-Objekten:
  - `CustomerAddressChanged`
    - `FirstName`: string
    - `LastName`: string
    - `Address`
      - Street: string
      - City: string
  - `UserLoginFailed`
    - `UserId`: int
    - `Reason`: string
- Objekte mit Zufallswerten befüllen (Random Strings)
- Objekte als JSON serialisieren
- Jeweils 10 Log-Einträge

Aufgabe 1

- Was beobachtet man in Kibana? („Launch Kibana“ im Stack-Menü)
  - Kann Kibana das Objekt serialisieren?
- Konkrete Aufgabe:
  - Suchen nach `UserLoginFailed` (type) nach einer spezielle `UserId` in Kabana
  - Kopieren Sie das Query + Result nach Fragen.txt

Antworten in Fragen.txt