# Lern- und Arbeitsauftrag *LA\_183\_17\_Logging.docx*

|  |  |
| --- | --- |
| Titel | Logging |
| Modul | 183 Informatiker/in EFZ |
| Autor / Version | Michael Schneider, Adrian Schmid / V2.0 |
| Hilfsmittel: | InsecureApp  PR\_183\_Logging.pptx  <https://learn.microsoft.com/de-ch/aspnet/core/fundamentals/logging> |
| Nachweis |  |
| Sozialform | Einzelarbeit / Partnerarbeit |
| Leistungsziele | 5.2, 5.3 |

## Ausgangslage

Das Logging in der BBB Insecure-App war bis zu dieser Version nicht wirklich durchdacht. Es wird gewünscht, dass die Applikation in dieser Hinsicht auf ein professionelles Niveau gebracht wird.

Ein Bild, das Blau, Screenshot, Majorelle Blue, Licht enthält.

Automatisch generierte Beschreibung

Quelle: <https://pixabay.com/illustrations/analytics-information-innovation-3088958/>, 03.10.2023

## Aufgabenstellung

### Teilaufgabe 1. Format einer Logzeile

Wie könnte eine Zeile aussehen für einen nicht erfolgreichen Loginversuch? Welchen «Level» würden Sie diesem Ereignis geben? Begründen Sie Ihre Wahl.

Beispiel: Timestamp: [Datum und Uhrzeit] - Level: Error - Benutzer: [Benutzername] - Nicht erfolgreicher Loginversuch

Der "Level" für dieses Ereignis sollte "Error" sein. Dies liegt daran, dass ein nicht erfolgreicher Loginversuch auf ein kritisches Ereignis hinweist, bei dem ein Benutzer möglicherweise Probleme beim Zugriff auf ein System hat. "Error" ist der passende Level, um solche schwerwiegenden Ereignisse zu kennzeichnen, da es sich um eine potenziell problematische Situation handelt, die auf Aufmerksamkeit und mögliche Fehlerbehebung hinweisen sollte.

### Teilaufgabe 2: Vorbereitung

Welche Ereignisse sollen geloggt werden mit welchem Level?

|  |  |
| --- | --- |
| **Ereignis** | **Level** |
| Erfolgreiches Login | Information |
| Fehlgeschlagenes Login | Warning |
| Passwort-Änderung | Information |
| Fehler bei 2 FA | Warning |
| News Create, Edit, Delete | Information |
| Versuch, News zu ändern die der Benutzer nicht darf | Warning |
| Versuch, News zu löschen die der Benutzer nicht darf | Warning |
| Ungültiges JWT-Token | Warning |
| Probleme mit Datenbank | Critical |

### ****Teilaufgabe 3: Implementation****

Implementieren Sie ihr Logging-Konzept aus Teilaufgabe 2. Verwenden Sie dafür die Logging Komponenten und Konzepte von ASP.Net Core. Um dieses Logging zu verwenden, muss es im Program.cs konfiguriert werden. Überall wo sie etwas Loggen wollen können Sie sich dann einen ILogger für die entsprechende Klasse injecten lassen.

## A screen shot of a computer program Description automatically generated

* var builder = WebApplication.CreateBuilder(args);: Ein WebApplication-Builder wird erstellt, um die Konfiguration und das Setup für die ASP.NET Core-Webanwendung vorzunehmen.
* builder.Logging.ClearProviders();: Die vorhandene Logging-Konfiguration wird gelöscht, um alle vorherigen Logger zu entfernen und sie neu zu konfigurieren.
* builder.Logging.AddConsole();: Ein Logging-Provider wird hinzugefügt, der Lognachrichten auf der Konsole ausgibt.
* builder.Logging.AddDebug();: Ein weiterer Logging-Provider wird hinzugefügt, der Lognachrichten für das Debugging bereitstellt.
* Weitere Provider können nach Bedarf hinzugefügt werden.
* var app = builder.Build();: Die Webanwendung wird erstellt und konfiguriert.

A screen shot of a computer program

Description automatically generated

In diesem Code wird ein privates Feld namens "\_logger" erstellt, das vom Typ "ILogger<UserController>" ist. Dieses Feld dient dazu, Logging-Funktionalität in der Klasse "UserController" zu verwenden.

Der Konstruktor für "UserController" nimmt einen Parameter vom gleichen Typ, "ILogger<UserController>", entgegen.

## Gütekriterien

Der Lern- und Arbeitsauftrag ist erfüllt, wenn …

* Sie ein Format einer Logzeile vorgeschlagen haben, welches alle wichtigen Punkte berücksichtigt.
* Sie die Ereignisse, die geloggt werden sollen, mit Level aufgelistet haben.
* Sie die BBB InsecureApp um ein professionelles Logging ergänzt haben.
* Sie das Logging für die BBB InsecureApp getestet haben.

## Zusätzliche Angaben zum Auftrag

Dokumentation: <https://learn.microsoft.com/de-ch/aspnet/core/fundamentals/logging>

**Code um das Loggin im Program.cs zu konfigurieren:**

builder.Host.ConfigureLogging(logging =>

{

logging.ClearProviders();

logging.AddConsole();

});

**Beispiel wie der Logger injected und verwendet werden kann:**

public class UserController : ControllerBase

{

private readonly ILogger \_logger;

public UserController(ILogger<UserController> logger)

{

\_logger = logger;

}

public ActionResult PasswordUpdate(PasswordUpdateDto request)

{

//...

\_logger.LogInformation("changing password of user {0}", user.Username);

//...

}

}