

Dokumentation - LogIt

Inhaltsverzeichnis

- Dokumentation - LogIt
 - Inhaltsverzeichnis
 - UML-Diagramme
 - Lastenheft
 - Zielsetzung
 - Muss-Kriterien
 - Kann-Kriterien
 - Pflichtenheft
 - Architektur
 - Datenbankmodell
 - API-Endpunkte (Auszug)
 - Softwarevoraussetzungen
 - Bedienungsanleitung
 - Installation
 - Backend (LogIt.Core)
 - Frontend (LogIt.UI)
 - Dokumentation
 - Nutzung
 - Screenshots
 - Tests
 - Backend (LogIt.Core)
 - Frontend (LogIt.UI)
 - Projekttagebuch
 - Quellen

UML-Diagramme

Diagramme mit PlantUML (<https://plantuml.com/de/>) erstellt

für Browseransicht code in PlantUML-Web-Editor (<https://plantuml.com/de/>) kopieren

- Klassendiagramm ([klassendiagramm.plantuml](#))



-
- Ablaufdiagramm (ablaufdiagramm.plantuml).



-

Lastenheft

Zielsetzung

Entwicklung eines Programms, das laufende Programme sowie deren Laufzeiten protokolliert und diese Daten grafisch sowie in einer Tabelle darstellt. Die Anwendung besteht aus einem Backend (ASP.NET Core) und einem Frontend (WPF).

Features

- Logging aller gestarteten Programme/Prozesse
- Speicherung der Sitzungen (Sessions) mit Start/Ende/Dauer
- API zur Verwaltung von Daten (REST, Swagger)
- Frontend mit grafischer Darstellung (Diagramm, Tabelle)
- Benutzerrollen (System, Backend, Frontend)
- Unit Tests und Logging
- Parallele Programmierung (Threads/Tasks)

Nice-to-have

- Mehr Details in der Tabelle (z.B. Prozess-ID, Pfad)
- Filter-/Suchfunktionen im GUI

- Einschaltbarer Start bei Systemstart
-

Pflichtenheft

Architektur

- **Backend (LogIt.Core):**
 - .NET Core WebAPI
 - Entity Framework Core mit SQL (lokal)
 - Serilog für Logging
 - API-Dokumentation via Swagger
 - Hintergrunddienst zur Prozessüberwachung
- **Frontend (LogIt.UI):**
 - WPF-Anwendung
 - LiveCharts2 für Diagramme
 - Kommunikation mit Backend über REST-API
 - Hauptfenster mit Tabellen- und Diagrammansicht

Datenbankmodell

- **Tabellen:** User, LogEntry, Session
- **Beziehungen:** 1 User → n LogEntries, 1 LogEntry → n Sessions
- **3. Normalform:** Alle Tabellen sind normalisiert

API-Endpunkte (Auszug)

- GET /api/logentries – Alle Programmeinträge
- GET /api/logentries/active – Aktive Programme
- POST /api/logentries – Neues Programm anlegen
- POST /api/logentries/{logId}/sessions – Neue Session anlegen

Softwarevoraussetzungen

- Visual Studio 2022
 - Windows 10/11
 - einen funktionierenden Computer
-

Bedienungsanleitung

Installation

Backend (LogIt.Core)

1. Visual Studio 2022 installieren
2. LogIt von Github klonen: [Github-Link \(https://github.com/Schokobaer06/LogIt/tree/main\)](https://github.com/Schokobaer06/LogIt/tree/main),
3. `LogIt.Core.sln` Im Pfad `./src/backend/LogIt.Core/` (<https://github.com/Schokobaer06/LogIt/tree/main/src/backend/LogIt.Core>) öffnen

Backend komplett eigenständig ohne Frontend ausführbar

LogIt.Core-Projekt in Visual Studio dafür öffnen und starten

Frontend (LogIt.UI)

1. Release von Github herunterladen: [Github-Link \(https://github.com/Schokobaer06/LogIt\)](https://github.com/Schokobaer06/LogIt),
2. ZIP-Archiv entpacken & öffnen
3. `LogIt.UI.exe` starten

Nach erstem Start wird eine Verknüpfung im Startmenü erstellt

Programm ab sofort auch über das Startmenü startbar

Dokumentation

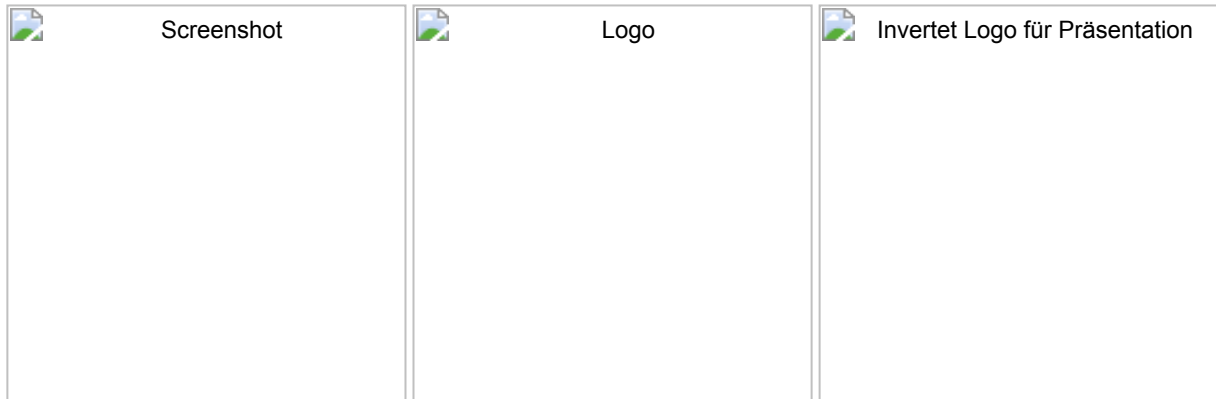
- API:
Swagger-Dokumentation unter [/API-Dokumentation \(API-Dokumentation/swagger.yaml\)](#), verfügbar
- Frontend:
Öffne [index.html \(Frontend-Dokumentation/html/index.html\)](#), im Verzeichnis `/Frontend-Dokumentation/html/`
- Backend:
Öffne [index.html \(Backend-Dokumentation/html/index.html\)](#), im Verzeichnis `/Backend-Dokumentation/html/`

Nutzung

- Programm starten und (optional im Hintergrund) laufen lassen

- Programme werden automatisch protokolliert und in der Tabelle & dem Diagramm angezeigt

Screenshots



Tests

Backend (LogIt.Core)

- **Unit Tests:** Für Models, Controller und Services (z.B. Datenbankzugriffe, Prozessüberwachung)
- **Integrationstests:** API-Endpunkte mit Testdaten
- **einige Tests:**
 - Anlegen eines neuen LogEntry
 - Starten/Beenden einer Session
 - Abfrage aller Sessions eines Programms
 - Fehlerfälle (z.B. ungültige IDs)

Frontend (LogIt.UI)

- **Unit Tests:** ViewModels (z.B. Datenaufbereitung, Formatierung)
- **Manuelle Tests:** UI-Bedienung, Diagrammaktualisierung, API-Kommunikation
- **einige Tests:**
 - Anzeige von Programmlisten und Sessions
 - Diagrammaktualisierung bei neuen Daten
 - Fehleranzeige bei Backend-Ausfall

Projekttagebuch

Datum	Beschreibung
2025-06-18	Updated Dokumentation

Datum	Beschreibung
2025-06-18	Updated Dokumentation & hinzugefügt: Swagger-Doku, Frontend/Backend-Doku
2025-06-17	Unit-Tests für Frontend & Backend hinzugefügt
2025-06-17	Dokumentation für Backend hinzugefügt
2025-06-17	Release 1.0 hinzugefügt
2025-06-17	Dokumentation & Kommentare zum Frontend-Code hinzugefügt
2025-06-16	app.ico & Code aktualisiert
2025-06-15	<code>.gitignore</code> um Zip-Exclude erweitert
2025-06-15	Veröffentlichung für Commit erstellt
2025-06-15	"End Präsentation" hinzugefügt
2025-06-15	Publish-Funktion hinzugefügt
2025-06-13	GUI aktualisiert
2025-06-13	UI aktualisiert
2025-06-13	Diagramm aktualisiert
2025-06-13	Graph funktioniert jetzt
2025-06-13	MainWindowViewModel + XAML aktualisiert
2025-06-13	LiveCharts2-Abhängigkeit hinzugefügt
2025-06-11	Automatisches Minimieren & Autostart hinzugefügt
2025-06-10	Graph hinzugefügt
2025-06-10	Frontend aktualisiert
2025-06-10	Anzeige von Programmnamen statt Prozessnamen implementiert
2025-06-06	Backend aktualisiert, um das Frontend nicht mitzuloggen
2025-06-06	Frontend startet Backend nun automatisch
2025-06-06	Funktionierendes Frontend hinzugefügt

Datum	Beschreibung
2025-06-04	Dauer-Berechnungsfehler behoben
2025-06-02	Backend fertiggestellt + Logik hinzugefügt
2025-05-28	ProcessMonitorService hinzugefügt
2025-05-28	Doku zu <code>LogItDbContext.cs</code> hinzugefügt
2025-05-28	REST-API + Datenbank hinzugefügt
2025-05-19	Klassendiagramm zur besseren Übersicht überarbeitet
2025-05-09	Alle Diagramme aktualisiert & ER-Diagramm hinzugefügt
2025-05-07	Erste PlantUML-Diagramme und VS Code-Settings hinzugefügt
2025-05-06	UML-Diagramm fertiggestellt
2025-05-05	UML + VS-Projektsetup hinzugefügt
2025-04-28	updated <code>.gitignore</code>
2025-04-28	Setup
2025-04-28	Initial Commit

Quellen

- **Diagramme:**
 - Erstellt mit PlantUML (<https://plantuml.com/de/>)
 - **Frameworks/Libraries:**
 - .NET 8.0 (<https://dotnet.microsoft.com/>)
 - Serilog (<https://serilog.net/>)
 - Entity Framework Core (<https://docs.microsoft.com/en-us/ef/core/>)
 - LiveCharts2 (<https://github.com/beto-rodriguez/LiveCharts2>)
 - Swashbuckle.AspNetCore (<https://github.com/domaindrivendev/Swashbuckle.AspNetCore>)
-

Stand: 18.06.2025