



Logistik-Services

Document type	<b>Work document</b>
Title	<b>Testaufgabe Applikationsentwickler/in .NET C#</b>
Number	1
Author	Julien Nyffeler und Andreas Nick
Contact details	julien.nyffeler@post.ch, andreas.nick@post.ch
Issued by	LS35.3
Validity	18.07.2024
Classification	A1
Version	V01.00
Issue date	7 April 2025
Replaces issue of	
Distribution list	

## 1. Allgemein

- Mit dieser Testaufgabe soll im Rahmen des Rekrutierungsprozesses bei LS35.3 eine kleine Applikation erstellt werden, welche in der zweiten Runde gemeinsam besprochen wird
- Die fachlichen Anforderungen sind bewusst einfach gehalten.
- Bei Unklarheiten dürfen Annahmen getroffen werden.
- Authentifizierung und Verschlüsselung sind nicht notwendig
- Bereitstellung: Der Source-Code + DB-Create-Script ist via E-Mail oder GitHub bereitzustellen
- Optional: Es soll ein lauffähiges Dockerfile geliefert werden.

## 2. Fachliche Anforderungen

Es soll ein Microservice erstellt werden, welcher Messdaten von Wetterstationen der Wasserschutzpolizei abrufen und diese in einer Datenbank persistent speichert.

Über die gespeicherten Messwerte sollen einfache Abfragen möglich sein.

Die Funktionen (Kapitel 2.2 und 2.3) des Microservice sind über eine REST/JSON-API bereitzustellen.

Eine Benutzeroberfläche ist nicht notwendig.

Im Fehlerfall soll eine Meldung zum aufgetretenen Fehler zurückgegeben werden.

### 2.1 Zu beziehende Attribute

Folgende Messwerttypen werden benötigt:

- Lufttemperatur
- Wassertemperatur
- Barometrischer Luftdruck
- Luftfeuchtigkeit
- 

Folgende Eigenschaften sind für die Weiterverarbeitung relevant:

- Messstation
- Messzeitpunkt
- Typ des Messwertes
- Wert
- Einheit

### 2.2 Bezug und Speicherung Wetterdaten

Die Wetterdaten sind über folgende Schnittstelle zu beziehen:

<https://opendata.swiss/de/dataset/messwerte-der-wetterstationen-der-wasserschutzpolizei-zurich2>

### 2.2.1 Parameter für Datenbezug

Folgende Wetterdaten sollen bezogen werden

- Daten der Messstationen «Tiefenbrunnen» und «Mythenquai»
- Datenzeitraum: Alle Daten vom Vortag
- Maximale Anzahl Daten pro Bezug: 100
  - o Bei mehr als 100 Einträgen sind die neusten 100 Einträge zu berücksichtigen

### 2.2.2 Filterung der Daten

Folgende Daten sind nicht zu verwenden

- Fehlerhafte Messungen
- Messungen ohne Messwert

### 2.2.3 Persistierung

Die benötigten Daten müssen persistent gespeichert werden.

### 2.2.4 Keine Duplikate

Mehrfach bezogene Messwerte dürfen nur einmal gespeichert werden

## 2.3 Abfrage von gespeicherten Wetterdaten

Gespeicherte Messdaten sollen abgefragt werden können.

### 2.3.1 Abfragemöglichkeiten

Für jeden Messwert-Typ sollen folgende Abfragen möglich sein

- Höchster gemessener Wert (je Einheit)
- Tiefster gemessener Wert (je Einheit)
- Durchschnittlich gemessener Wert (je Einheit)
- Anzahl gespeicherte Messwerte (je Einheit)
- Bezug aller Messwerte (alle vorhandenen Attribute)

### 2.3.2 Einschränkungen

Für jede Abfrage sollen die folgenden Einschränkungen möglich sein

- Herkunft Messstation (Optional)
- Start und End-Zeitpunkt der Messung (Pflicht)

### 2.3.3 Rückgabe

Bei der Abfrage nach dem höchsten Wert, nach dem tiefsten Wert und dem Bezug aller Messwerte, sollen immer die ganzen Datensets zurückgegeben werden, also mit allen Elementen, welche in Kapitel 2.1 Zu beziehende Attribute definiert sind.

## 3. Technische Anforderungen

- Typ der Applikation: Microservice
- Technologie: C# / .Net 9
- Bereitzustellende Schnittstelle: REST/API
- RDS: Datenbank (MSSQL, Postgres oder SQLite)

## 4. Testing / Documentation

- Die Funktionen des Microservice sollen mittels Unit-Tests getestet sein
- Die Code-Bestandteile sollen minimal dokumentiert sein