

Wetr.Web

Voraussetzungen für erfolgreiche Inbetriebnahme

- npm
- Angular CLI
- Wetr.WebService läuft mit Port '54405', ansonsten unter /src/app/services/api/api-configuration.ts anpassen
- MySql Docker Container aktiv, damit Wetr.WebServer einwandfrei funktioniert

Services

Die API Services wurden am Anfang mithilfe der swagger.json Datei generiert. Danach wurden die generierten Services per Hand angepasst. Die root-Url der API kann in der Datei api-configuration.ts angepasst werden. Neben den API-Services existiert noch ein Authentication-Service (für die Authentifizierung) und Dashboard-Service (für das persistieren der Benutzerpräferenzen).

API Services

Base

Stellt die Basis aller API Services. Enthält einige Hilfsmethoden wie z.B. `newParams(): HttpParams`.

Station Service

- `GetAll(): Observable<Array<Station>>` Holt sich alle Stationen.
- `GetByUserName(userName: string): Observable<Array<Station>>` Holt sich nur die Stationen für den mitgegeben Benutzernamen.
- `GetById(id: number): Observable<Station>` Holt sich nur eine Station mit gegebener Id.
- `GetAllStationTypes(): Observable<Array<StationType>>` Liefert alle Typen von Stationen.
- `GetAllCommunities(): Observable<Array<Community>>` Liefert alle Gemeinden.
- `UpdateStation(s: Station): Observable<null>` Aktualisiert eine Station.
- `InsertStation(s: Station): Observable<null>` Fügt eine neue Station hinzu.
- `DeleteStation(s: Station): Observable<null>` Löscht eine Station.

Measurement Service

- `ByFilter(filterSettings: FilterSettings): Observable<Array<Measurement>>` Holt sich alle Messungen mit gegeben Filter. Der Filter beinhaltet ein Feld von Station, eine Zeitspanne (von/bis), ein Intervalltyp (Tag/Week/Month/Year) und ein Akkumulierungstyp (Min/Max/Avg).

- `ByStations(stations:Array<Station>, from:Date, to:Date):Observable<Array<Measurement>>` Eine einfache Abfrage für ein Feld von Messungen.
- `Insert(m: Measurement): Observable<null>` Fügt eine Messung hinzu.

User Service

- `Login(username:string, password:string): Observable<boolean>` Ruft die API-Route für den Login auf. Die Überprüfung erfolgt im Authentication-Service.

Search Service

- `BySearchValue(searchValue: string): Observable<Array<SearchResult>>` Liefert aus einem String die dazugehörigen Suchresultate.
- `BySearchResults(searchResults: Array<SearchResult>): Observable<Array<Station>>` Liefert aus einem Feld von Suchresultaten die dazugehörigen Stationen.

Authentication Service

- `public login(username:string, password:string): Observable<boolean>` Hier erfolgt die Überprüfung auf ein erfolgreiches Login. Wenn es erfolgreich war, wird im lokalen Speicher ein Item (`currentUser`) angelegt. Der Wert dieses Attributes ist aktuell der Benutzername, sollte aber mit einem JSON Web Token ausgetauscht werden.
- `public getCurrentUser(): string` Liefert den Benutzername, welcher im lokalen Speicher gespeichert wird.
- `public isLoggedIn(): boolean` Überprüft ob im lokalen Speicher ein Schlüssel mit 'currentUser' existiert und liefert den boolschen Wert der Überprüfung zurück.
- `public logout()` Löscht das Item (key: 'currentUser') im lokalen Speicher.

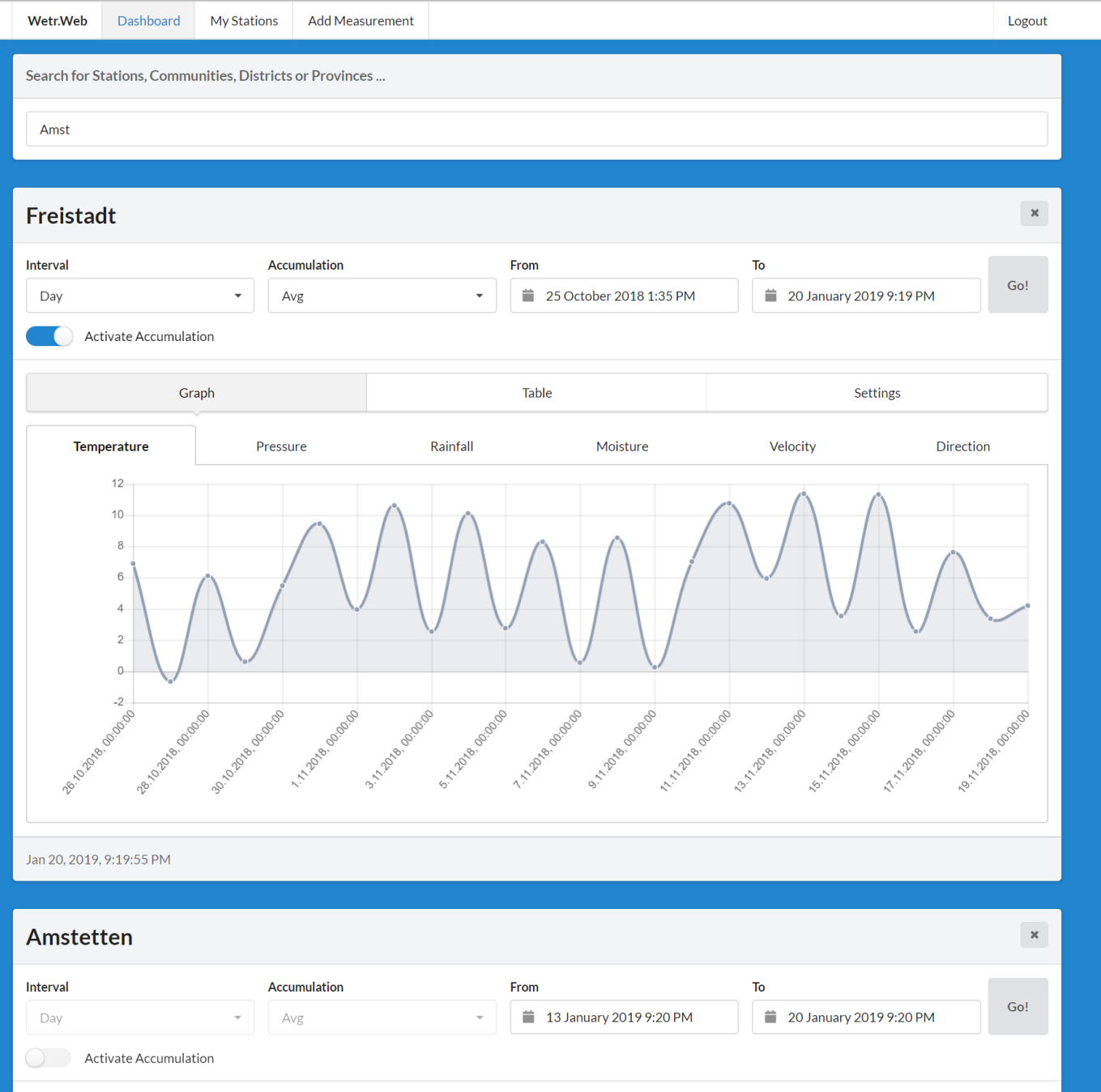
Dashboard Service

- `public.getItems():DashboardItem[]` Liefert alle, im lokalen Speicher, gespeicherten Dashboard-Items.
- `public saveItem(item:DashboardItem)` Persistiert ein Dashboard-Item im lokalen Speicher.
- `public removeItem(item:DashboardItem)` Löscht ein Dashboard-Item aus dem lokalen Speicher.
- `public updateItem(item:DashboardItem)` Aktualisiert ein Item im lokalen Speicher.

Komponenten

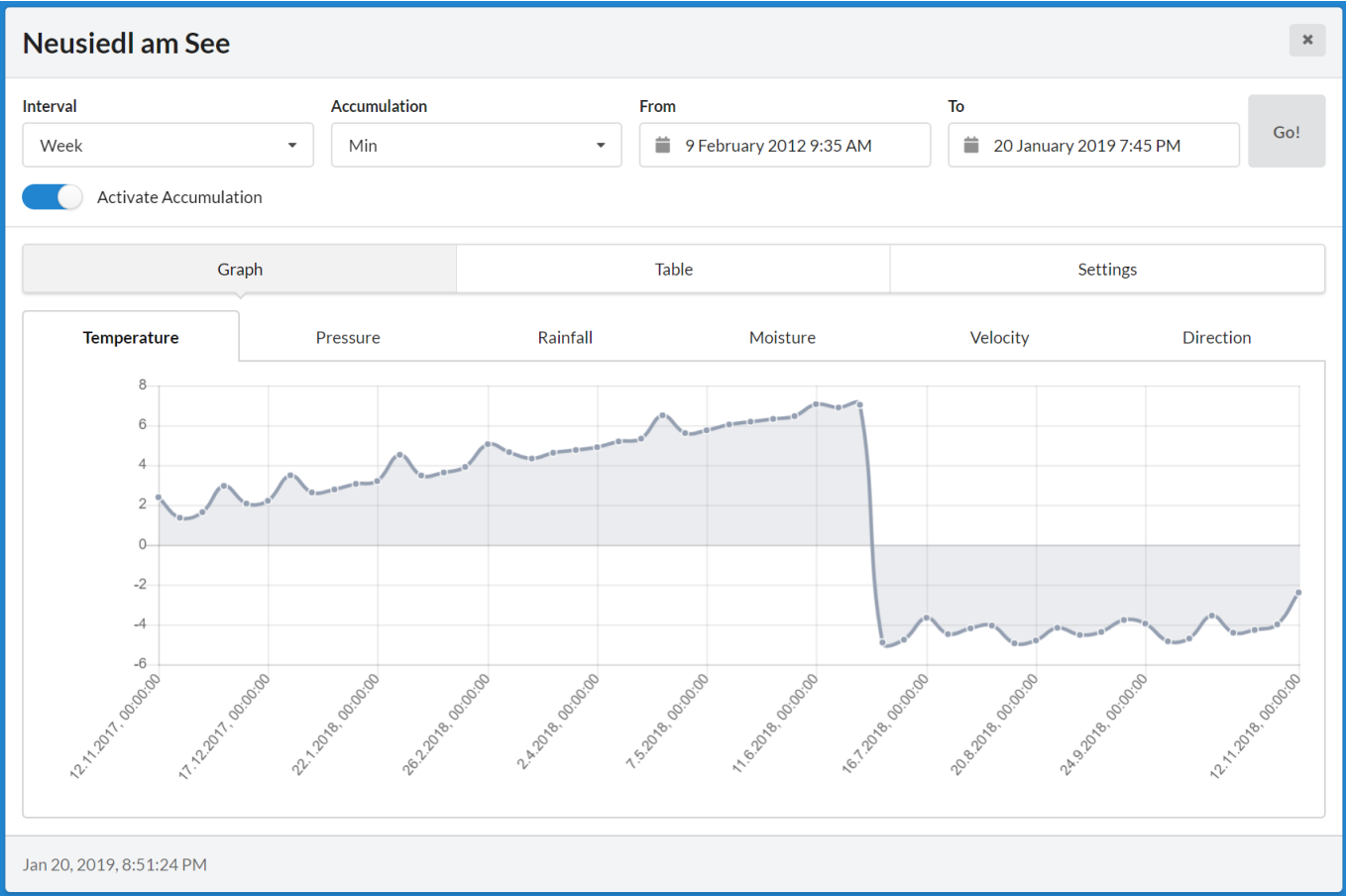
DashboardComponent

Zeigt das Dashboard an. Holt sich die Dashboard-Items mithilfe des Services und generiert daraus DashboardItemComponents.



DashboardItemComponent

Ist eine einzelne Anzeige auf dem Dashboard. Hier können verschiedenste Abfragen durchgeführt und die Präferenzen gespeichert werden.



DashboardItemSettingsComponent

Ist eine Teilkomponente eines Dashboard-Items. Hier kann die Konvertierung der Temperatur (Celsius, Fahrenheit, ...) eingestellt werden.

Temperature conversion

☒ Celsius (°C)

☐ Fahrenheit (°F)

☐ Kelvin (K)

☐ Rankine (°R)

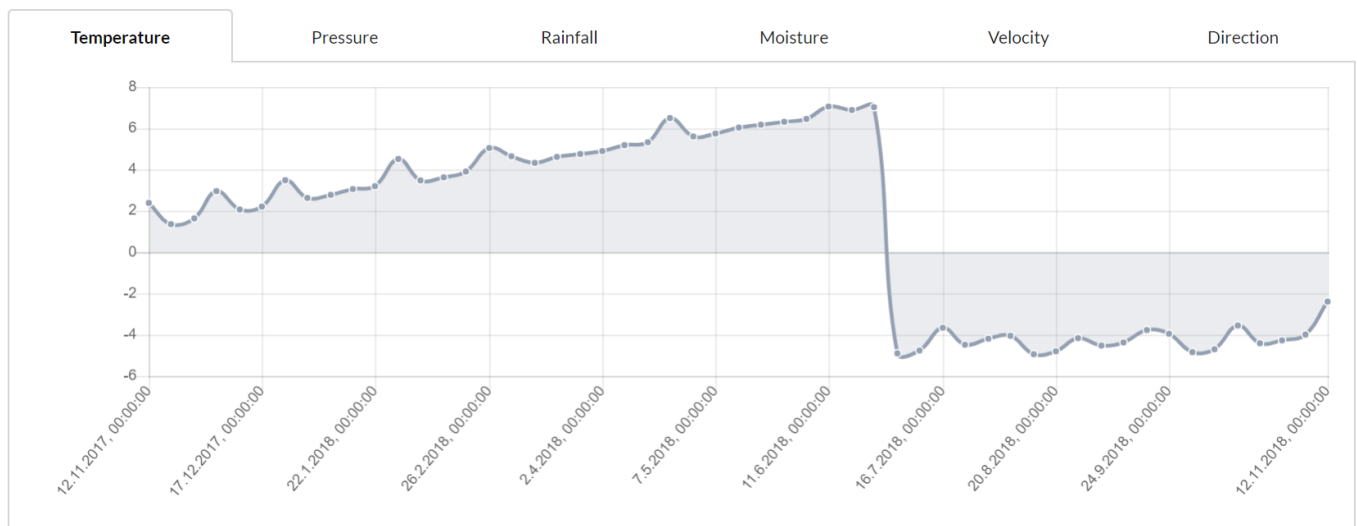
☐ Delisle (°De)

☐ Réaumur (°Ré)

☐ Rømer (°Rø)

ChartComponent

Zuständig für die Anzeige als Diagramm. Enthält auch ein kleines Menü, um durch die verschiedenen Messtypen zu navigieren (Temperature, Pressure, ...).



LoginComponent

Kümmert sich um die Eingabe der Login-Daten und um entsprechende Weiterleitung bei Erfolg.

Wetr.Web Login

Username

Password

Login

MeasurementFormComponent


Zuständig für die valide Eingabe von Messdaten. Damit können neue Messungen hinzugefügt werden.

New Measurement

Station

1 - Andau1 ▼

Date

 20 January 2019 9:18 PM

Temperature (in °C)

0

Pressure

0

Rainfall

0

Moisture

0

Velocity

0

Direction

0

Save

MeasurementTableComponent


Ist wie ChartComponent eine Anzeigekomponente für ein Dashboard-Item.

DateTime	Temperature	Pressure	Rainfall	Moisture	Velocity	Direction
2017-11-12T00:00:00	2.39	1024.08	0	0.46	7	0
2017-11-19T00:00:00	1.36	1023.66	0	0.29	6	0
2017-11-26T00:00:00	1.64	1023.79	0	0.28	7	0
2017-12-03T00:00:00	2.96	1023.68	0	0.28	7	0
2017-12-10T00:00:00	2.07	1023.7	0	0.39	7	0
2017-12-17T00:00:00	2.21	1023.74	0	0.36	7	0
2017-12-24T00:00:00	3.49	1023.76	0	0.24	7	0
2018-01-01T00:00:00	2.63	1023.77	0	0.23	8	0

MyStationsListComponent

Liefert eine Liste mit den eigenen Stationen. Nur möglich wenn man eingeloggt ist, ansonsten wird man auf die Login-Seite navigiert.

Wetr.Web	Dashboard	My Stations	Add Measurement	Logout
----------	-----------	-------------	-----------------	--------

<div> Add Station</div>	<div>Andau1</div> <div>Typ 1 - Marke 1 Type</div> <div>Andau 7163</div> <div>Neusiedl am See, Burgenland</div> <div>47.77249908, 17.03333282 Coordinates</div> <div>DeleteEditDetails</div>	<div>Bernstein</div> <div>Typ 1 - Marke 1 Type</div> <div>Bernstein 7434</div> <div>Oberwart, Burgenland</div> <div>47.40833282, 16.26138878 Coordinates</div> <div>DeleteEditDetails</div>
<div>Bruckneudorf</div> <div>Typ 1 - Marke 1 Type</div> <div>Bruckneudorf 2460</div> <div>Neusiedl am See, Burgenland</div> <div>48.01277924, 16.84499931 Coordinates</div> <div>DeleteEditDetails</div>	<div>Güssing</div> <div>Typ 1 - Marke 1 Type</div> <div>Güssing 7540</div> <div>Güssing, Burgenland</div> <div>47.0625, 16.32194328 Coordinates</div> <div>DeleteEditDetails</div>	<div>Lutzmannsburg</div> <div>Typ 1 - Marke 1 Type</div> <div>Lutzmannsburg 7361</div> <div>Oberpullendorf, Burgenland</div> <div>47.46527863, 16.64555555 Coordinates</div> <div>DeleteEditDetails</div>
<div>Mattersburg</div> <div>Typ 1 - Marke 1 Type</div>	<div>Neusiedl Am See</div> <div>Typ 1 - Marke 1 Type</div>	<div>Wörterberg</div> <div>Typ 1 - Marke 1 Type</div>

MyStationsListItemComponent

Ein Item für die Liste der eigenen Stationen. Zeigt Informationen zu einer Station an. Außerdem gibt es die Möglichkeit Stationen zu löschen oder auf die Details/Editierungs-Seite zu navigieren.

Andau1	
Typ 1 - Marke 1 Type	
Andau 7163	Neusiedl am See, Burgenland
47.77249908, 17.03333282 Coordinates	
<div>DeleteEditDetails</div>	

MenuComponent

Das Hauptmenü, welches immer (außer bei Login) angezeigt wird. Benötigt um zu Navigieren.

Wetr:Web	Dashboard	My Stations	Add Measurement		Logout
----------	-----------	-------------	-----------------	--	--------

SearchComponent

Für die Suche von Stationen, Gemeinden, Bezirken oder Ländern (Provinzen) zuständig. Befindet sich am Anfang auf der Dashboard-Seite. Bei Eingabe wird ein Dropdown-Menü geöffnet mit Suchvorschlägen.

Search for Stations, Communities, Districts or Provinces ...

Search for stations ...

SearchItemComponent

Stellt einen Suchvorschlag für die Suche dar.

Search for Stations, Communities, Districts or Provinces ...

Frei

Community
Freistadt

District
Freistadt

Station
Freistadt

StationDetailsComponent

Zeigt die Details einer Station an. Wenn man eingeloggt und der Besitzer der jeweiligen Station ist, werden auch Lösch- und Editierfunktionen freigeschaltet.

Andau1	
admin Owner	Typ 1 - Marke 1 Type
Andau 7163	Neusiedl am See, Burgenland
47.77249908, 17.03333282 Coordinates	
<div>DeleteEditDetails</div>	

StationFormComponent

Für die Eingabe von Stationsdaten zuständig. Damit können bereits existierende Stationen editiert und neue hinzugefügt werden.

Edit Station

Name

Andau1

Type of Station

Typ 1 - Marke 1 ▼

Street

test2131

Community

7163 Andau, Neusiedl am See, Burgenland ▼

Save

New Station

Name

Type of Station

Street

Community

Web

This project was generated with [Angular CLI](#) version 6.1.3.

Development server

Run `ng serve` for a dev server. Navigate to `http://localhost:4200/`. The app will automatically reload if you change any of the source files.

Code scaffolding

Run `ng generate component component-name` to generate a new component. You can also use `ng generate directive|pipe|service|class|guard|interface|enum|module`.

Build

Run `ng build` to build the project. The build artifacts will be stored in the `dist/` directory. Use the `--prod` flag for a production build.

Running unit tests

Run `ng test` to execute the unit tests via [Karma](#).

Running end-to-end tests

Run `ng e2e` to execute the end-to-end tests via [Protractor](#).

Further help

To get more help on the Angular CLI use `ng help` or go check out the [Angular CLI README](#).