



FH MÜNSTER  
University of Applied Sciences

Department of Electrical Engineering and Computer Science

# ANGULAR WEB-APP ZUR MESSDATENVISUALISIERUNG

MASTERPROJEKT

*Author:*  
Robin Weiß

*Betreuer:*  
Prof. Dr.-Ing. Peter Glösekötter

April 2017

# INHALTSVERZEICHNIS

---

1	EINLEITUNG	1
1.1	Anforderungen	1
2	IMPLEMENTIERUNG	3
2.1	Domain Model	3
2.2	Frameworks und Technologien	3
2.2.1	Angular	3
2.2.2	ASP.NET Core	3
2.2.3	Entity Framework Core	5
2.2.4	SQLite	5
2.2.5	Bootstrap und Now UI Kit	5
2.2.6	Highcharts	6
2.3	JSON-Fileformat	6
2.4	REST API	7
2.5	Zugriffsrechte	8
2.6	Funktionalitäten	8
3	ZUSAMMENFASSUNG	13
3.1	Fazit	13
3.2	Ausblick	13
	LITERATUR	14

## ABBILDUNGSVERZEICHNIS

---

Abbildung 1	Domain Model als UML Klassenhierarchie	4
Abbildung 2	Upload-View	8
Abbildung 3	Datenbank View mit Messungs-Ansicht	9
Abbildung 4	Datenbank View mit Stack-Ansicht	9
Abbildung 5	Stack-Detail-Ansicht	10
Abbildung 6	Stackmessung	10
Abbildung 7	Zoom-Funktion vorher	11
Abbildung 8	Zoom-Funktion nachher	11
Abbildung 9	Ladungsplot aktiviert nachher	11
Abbildung 10	Ladungsplot deaktiviert	12
Abbildung 11	Detailansicht einer Ortskurve	12
Abbildung 12	Notifications	12

## TABELLENVERZEICHNIS

---

## LISTINGS

---

Listing 1	JSON Fileformat Ortskurven	6
Listing 2	JSON Fileformat Zeitreihen	7

## EINLEITUNG

---

Der steigenden Bedarf der Speicherung elektrischer Energie erfordert die stetige Weiterentwicklung von Batteriespeichersystemen. Die elektrochemische Reaktion von Zink mit Luftsauerstoff bietet ein beträchtliches Potenzial für wiederaufladbare Speicher für große Energiemengen, bei gleichzeitig hohen Energiedichten. Der besondere Vorteil von Zink-Luft-Akkumulatoren liegt neben der theoretisch dreifach größeren Energiedichte als der von Lithium-Ionen-Akkumulatoren in den niedrigen Kosten und der hohen Sicherheit.

Im Rahmen eines Verbundforschungsprojektes der Technischen Universität Dortmund und der Fachhochschule Münster wurde ein neuartiger Zink-Luft-Akku mit dem dazugehörigen Batterie-Management-System entwickelt. Um die bei der Entwicklung anfallenden Messdaten für zukünftige Auswertungen zentral zu speichern und sie anderen Projektpartnern zur Verfügung zu stellen, wurde im Rahmen eines Masterprojektes eine Web-App entwickelt, die die aus den unterschiedlichen Versuchsreihen anfallenden Messdaten visualisiert und zur weiteren Verarbeitung zum Download zur Verfügung stellt.

### 1.1 ANFORDERUNGEN

Nachfolgend werden die Anforderungen an die Web-App im Detail beschrieben:

- Messdaten sollten in einem einheitlichen Format hochladbar sein und automatisch in eine Datenbank übernommen werden. Metadaten und Zusatzinformationen zu den jeweiligen Messaufbauten und -abläufen sollten in den jeweiligen Files gespeichert und automatisch ausgelesen werden können.
- Die als Zeitreihen und Ortskurven vorliegenden Messdaten sollten visualisiert werden. Dabei sollten unterschiedliche Ortskurven und Zeitreihen eines Lade- bzw. Entladevorgangs gemeinsam dargestellt werden können. Das zeitliche Intervall zur Darstellung einer Messreihe sollte veränderbar sein. Um eine ausreichend hohe Auflösung bei gleichzeitig optimiertem HTTP-Traffic zu erreichen, sollten Daten beim Zoomen "lazy" nachgeladen werden.
- Um Zukunftssicherheit und Skalierbarkeit zu gewährleisten, sollten zusätzlich zu einzelnen Zellen auch Stacks, das sind Verschaltungen von einzelnen Zellen, und Systeme, das sind Verschaltungen von einzelnen Stacks, angelegt werden können. Die

App sollte eine Filterfunktion nach unterschiedlichen Kategorien haben und eine Suchfunktion in der jeweiligen Kategorie.

- Um zu verhindern, dass Dritte die gespeicherten Daten einsehen und manipulieren können, sollte der Zugriff auf die Web-App über einen Login mit gemeinsamen Passwort geschützt werden.
- Das User Interface sollte responsive sein, damit die App auch auf mobilen Geräten nutzbar ist.

## IMPLEMENTIERUNG

---

Um die entwickelte Anwendung plattformübergreifend einsetzen zu können, wurde eine Angular-Single-Page-App mit ASP.NET-Back-End entwickelt. Dieses Kapitel beschreibt die Implementierung und die dabei genutzten Komponenten im Detail.

### 2.1 DOMAIN MODEL

Das Domain Model besteht aus den in Abbildung 7 dargestellten Klassen. Dabei wird eine Messung durch Messdaten sowie zugehörige Metadaten wie Name und Beschreibung repräsentiert. Eine Messung kann entweder zu einer Zelle, einem Stack oder einem System gehören. Ein Stack ist eine Verschaltung von mehreren Zellen, ein System eine Verschaltung von mehreren Stacks. Die Enumeration `MeasurementType` beschreibt den Typ der Messung: undefiniert, Zeitreihe, Ortskurve, oder Sonstige. `CircuitType` gibt die Schaltungsart an: Reihen- oder Parallelschaltung. Neue Instanzen des jeweiligen Typs können zur Datenbank hinzugefügt, gelöscht, und bearbeitet werden. Es stehen jeweils List- und Detail-Views zur Verfügung. Die Messung hat eine Upload- und eine Download-Methode. Systeme werden intern als Batteries bezeichnet, um Konflikte mit C# Bibliotheken zu vermeiden.

### 2.2 FRAMEWORKS UND TECHNOLOGIEN

Im Folgenden werden die verwendeten Frameworks und Technologien näher beschrieben.

#### 2.2.1 *Angular*

Angular 4.x [3] ist ein TypeScript-basiertes Frontend-Webframework. Angeführt durch Google, wird es von einer Community aus Einzelpersonen und Unternehmen entwickelt und als Open-Source-Software über GitHub [4] publiziert.

#### 2.2.2 *ASP.NET Core*

Die General-Purpose-Development-Plattform .NET Core wird unter der Koordination von Microsoft zusammen mit der .NET Community entwickelt und ist als Open Source-Projekt über GitHub [7] ver-

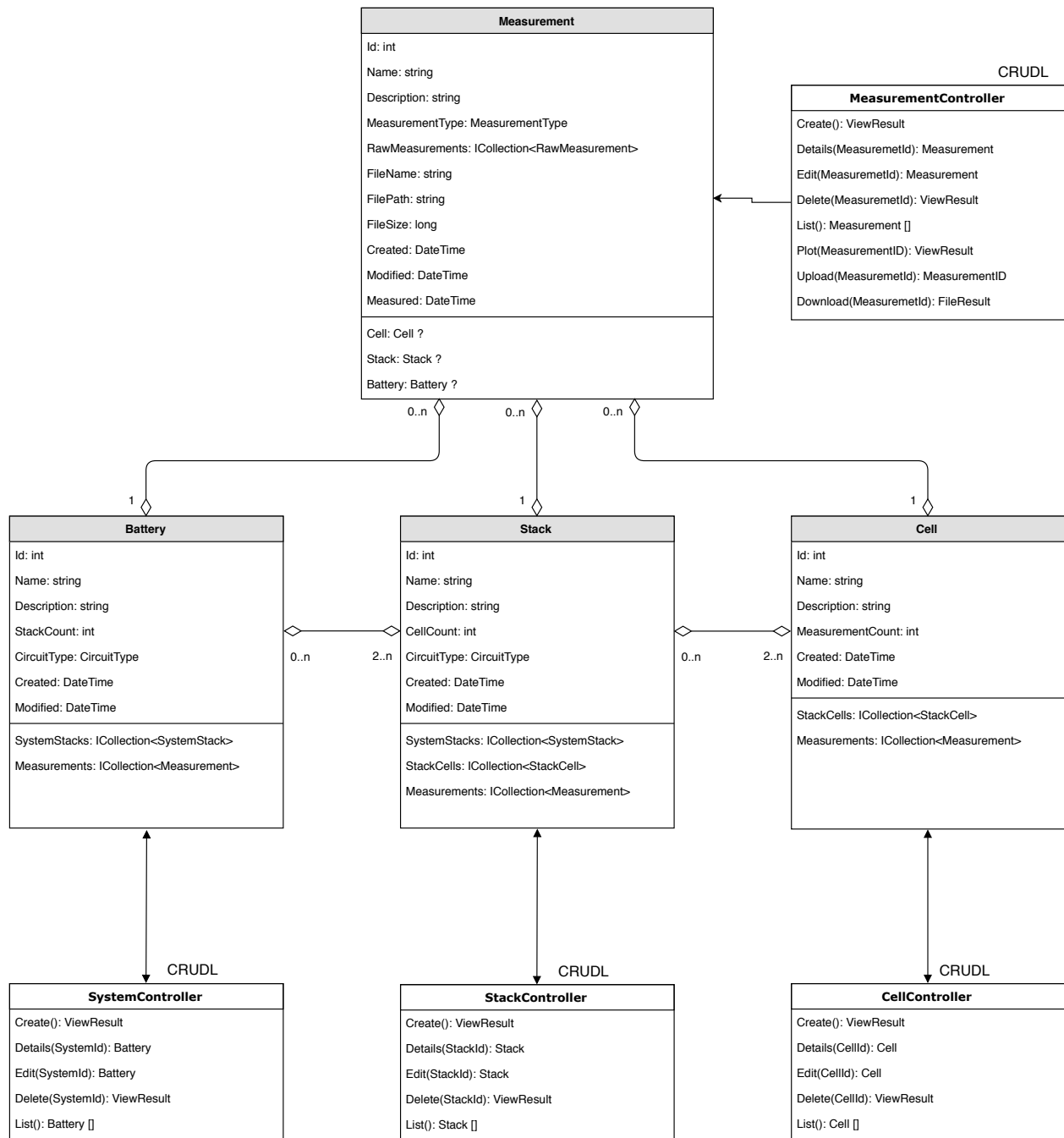


Abbildung 1: Domain Model als UML Klassenhierarchie

füßbar. Sie bietet eine plattformübergreifende Lösung zur Entwicklung und Ausführung von Anwendungsprogrammen und läuft unter Windows, macOS und Linux. ASP.NET steht für Active Server Pages .NET und ist ein Web Application Framework von Microsoft, mit dem sich dynamische Webseiten, Webanwendungen und Webservices in C# entwickeln lassen. Im Hintergrund läuft der Kestrel Web Server für ASP.NET [1]. Als Entwicklungsumgebung wurde Visual Studio Code genutzt, ein kostenloser und quelloffener Texteditor, mit C#-Extension.

### 2.2.3 *Entity Framework Core*

Entity Framework Core ist eine abgespeckte, erweiterbare, plattformübergreifende Version des bekannten Entity Frameworks, ein Framework für objektrelationale Abbildung von Microsoft. Entity Framework Core ermöglicht es ausschließlich unter Benutzung von .NET-Objekten mit Datenbanken zu arbeiten und beseitigt so die Notwendigkeit eigenen Code zum Zugriff auf die Datenbank zu schreiben [8]. EF Core verwendet das aus Entity-Klassen bestehende C#-Domain-Model um daraus mit Hilfe von Migrations eine Datenbank zu erstellen. So kann das Domain Model während der Entwicklung einfach erweitert oder geändert werden, ohne händisch SQL-Befehle schreiben und testen zu müssen. Ein Datenzugriff auf eine Instanz einer Entity-Klasse wird durch Language Integrated Queries (LINQ) beschrieben [7].

### 2.2.4 *SQLite*

SQLite implementiert eine eigenständige, serverlose SQL-Database-Engine und ist dabei die weltweit meistverwendete ihrer Art [6]. Anders als die meisten anderen SQL Datenbanken schreibt und liest SQLite direkt in und aus herkömmlichen Files auf einem Datenträger. Dabei ist das Fileformat plattformübergreifend und frei kopierbar zwischen 32-bit und 64-bit Systemen sowie zwischen Big-Endian und Little-Endian Architekturen.

### 2.2.5 *Bootstrap und Now UI Kit*

Bootstrap [9] ist ein freies CSS-Framework das responsive Webdesign vereinfacht. Es enthält auf HTML und CSS basierende Gestaltungsvorlagen für Typografie, Formulare, Buttons, Tabellen, Grid-Systeme, Navigations- und andere Oberflächengestaltungselemente. Bootstrap ist modular aufgebaut und besteht im Kern aus Less-Stylesheets. Als Erweiterung wurde das Now UI Kit [2] verwendet, was ein modernes, flaches und intuitives User Interface realisiert.



### 2.2.6 Highcharts

Highcharts [5] ist eine in pure JavaScript geschriebene Visualisierungs-library der Firma Highsoft. Mit ihr lassen sich unter anderem interaktive Lineplots und Ortskurven erstellen. Auch lazy loading von Daten wird mit ihr vereinfacht.

### 2.3 JSON-FILEFORMAT

Das JSON-Format wurde gewählt, weil es in C# vorgefertigte JSON-to-Object Parser gibt und es flexibel und beliebig erweiterbar ist. Anders als beim ursprünglichen zum Logging genutzten CSV-Format, ist es einfach Metadaten und Zusatzinformationen zu speichern, ohne diese fest in gewissen Zeilen und Spalten zu kodieren. Weiterhin ist es menschenlesbar und portabel, anders als .mat-Files, die ursprünglich zur Speicherung der Ortskurven dienten. Listing 1 und 2 zeigen jeweils die typische Struktur eines JSON-Files für eine Ortskurve und eine Zeitreihe mit beschreibenden Kommentaren.

Listing 1: JSON Fileformat Ortskurven

```

1 {
2     // 0 = Undefined, 1 = Zeitreihe,
3     // 2 = Ortskurve, 3 = Sonstige
4     "type": 2,
5     "battery": null, // System ID
6     "stack": 3, // Stack ID
7     "cell": null, // Zellen ID
8     "time": 1502702593000, // UNIX Timestamp
9     "mode": "Discharge", // State of Charge: Laden oder Entladen
10    "current": -1000, // Ladestrom
11    "charge": -308843203.7, // Ladung
12    "spectrum": {
13        "frequency": [ // Frequenzen
14            10,
15            20,
16            30,
17            40,
18            50
19        ],
20        "impedance": {
21            "_ArrayType_": "double",
22            "_ArraySize_": [37, 1],
23            "_ArrayIsComplex_": 1,
24            "_ArrayData_": [ // Komplexe Impedanzen
25                [0.09263688244, 0.01274971625],
26                [0.09890474907, -0.00793708269],
27                [0.08428400943, -0.009960039118],
28                [0.08048473845, -0.01711342423],
29                [0.08200628526, -0.0163586119]

```

```

30         ]
31     },
32 },
33 "rawdata": [{
34     "frequency": 0.1,
35     "timepoints": [ // Zeitstempel
36         0,
37         5.243e-06,
38         1.0486e-05,
39         1.5729e-05,
40         2.0972e-05,
41         2.6214e-05,
42         3.1457e-05
43     ]
44 }]
45 }

```

Listing 2: JSON Fileformat Zeitreihen

```

1 {
2     "measurementType": 1, // 0 = Undefined, 1 = Zeitreihe, 2 = Ortskurve, 3 = Sonstige
3     "stackId": 121,
4     "data": [
5         {
6             "capacity": -13970,
7             "current": 0,
8             "mode": "discharging",
9             "time": 1505127209192,
10            "voltage": 1378
11        },
12        {
13            "capacity": -25217,
14            "current": -1419,
15            "mode": "discharging",
16            "time": 1505127219195,
17            "voltage": 1106
18        },
19        {
20            "capacity": -43090,
21            "current": -1861,
22            "mode": "charging",
23            "time": 1505127229189,
24            "voltage": 1100
25        }
26    ]
27 }







```

## 2.4 REST API




Representational State Transfer (REST) bezeichnet ein Programmierparadigma für verteilte Systeme, insbesondere Webservices. In einer


## Upload

2 Dateien ausgewählt:

Name	Größe	Fortschritt	Status	Aktionen
Ortskurve.json	19.857 MB	<div><div></div></div> 100%	✓	  
Zeitreihe.json	11.858 MB	<div><div></div></div> 51%		  

Gesamtfortschritt:  66%

 Upload all  Cancel all  Remove all



ⓘ Drag files here or click to browse.

Abbildung 2: Upload-View

Client-Server-Architektur stellt der Server einen Dienst bereit, der bei Bedarf vom Client angefragt werden kann. So stellt die REST API in unserer Single-Page-App Mess- und Metadaten aus der Datenbank bereit, die bei Bedarf dynamisch geladen bzw. nachgeladen werden.

## 2.5 ZUGRIFFSRECHTE

Der Zugriff wurde durch eine .htaccess-Datei Konfigurationsdatei beschränkt, sodass nur Benutzer mit einem gemeinsamen Passwort Daten einsehen, hochladen, herunterladen und verändern können.

## 2.6 FUNKTIONALITÄTEN

Nachfolgend ist ein möglicher Workflow beschrieben um die Funktionalitäten der App zu erläutern.

Klickt man über die Navigationsleiste am obigen Bildschirmrand auf Upload, gelangt man zum Upload-View, wie in Abbildung 2 dargestellt, mit Hilfe dessen man per Drag-and-drop oder per Filebrowser Messdatenfiles zum Hochladen auswählen kann. Die Files werden in einer Warteschlange mit der jeweiligen Filegröße angezeigt und können einzeln oder alle nacheinander hochgeladen werden. Eine Progress-Bar zeigt jeweils Einzel- und Gesamtfortschritt des Uploads an.

War der Upload erfolgreich wird man zur Datenbank weitergeleitet, in der, wie in Abbildung 3 zu sehen ist, durch Klicken auf die Seitenleiste eine Ansicht nach Systemen, Stacks, Zellen und Messungen ausgewählt werden kann. Die Ansicht mit den Messungen wird nach dem Upload automatisch ausgewählt. Hier finden sich alle Mes-

ID	Name	Typ	Größe	Geändert	Aktionen
14	Zeitreihe.json	ZEITREIHE	11.858 MB	Oct 15, 2017	[i] [p] [x]
13	Testmessung Ortskurve	ORTSKURVE	19.857 MB	Oct 15, 2017	[i] [p] [x]
12	test_25_09_17.json	ZEITREIHE	10.591 MB	Oct 14, 2017	[i] [p] [x]
11	test_22_09_17.json	ZEITREIHE	4.428 MB	Oct 14, 2017	[i] [p] [x]
10	test_20_09_17.json	ZEITREIHE	3.044 MB	Oct 14, 2017	[i] [p] [x]
9	test_19_09_17.json	ZEITREIHE	0.022 MB	Oct 14, 2017	[i] [p] [x]

Abbildung 3: Datenbank View mit Messungs-Ansicht

Suche

- Systeme
- Stacks**
- Zellen
- Messungen

+ Neuer Stack

Stack ID: 4

Test Stack 4

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquid ex ea commodo consequat. Quis aute iure reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint obcaecat cupiditat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

[i] [p] [x]

Erstellt: Aug 15, 2017, 9:03 PM      Geändert: Aug 15, 2017, 9:03 PM

Stack ID: 3

Test Stack 3

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed eiusmod tempor incididunt ut labore et

Abbildung 4: Datenbank View mit Stack-Ansicht

sungen in der Datenbank, sortiert in absteigender Reihenfolge, also mit den letzten Uploads ganz oben. Zusätzlich zum Namen der Messung, der mit dem Dateinamen initialisiert wird wenn es nicht anders im JSON-File spezifiziert ist, sind der Typ der Messung, also Zeitreihe oder Ortskurve, die Größe und das Datum der letzten Änderung in einer Tabelle dargestellt. Über drei Aktionsbuttons kann man die Details der Messung einsehen, sie bearbeiten oder löschen. Es werden jeweils 10 Messungen pro Seite dargestellt.

Wählt man beispielsweise die Ansicht Stacks, wie in Abbildung 4 dargestellt, so werden die Stacks kachelweise mit Zusatzinformationen wie der Art der Verschaltung – zum Beispiel Reihenschaltung – und der Anzahl der zugehörigen Zellen angezeigt. Außerdem wird der Beschreibungstext und das Erstellungs- sowie letzte Änderungsdatum angezeigt. Über die oben beschriebenen Aktionsbuttons kann der Stack geändert oder gelöscht werden. Klickt man auf Details, gelangt man in die Detailansicht.

Die Detailansicht in Abbildung 5 zeigt wiederum die Beschreibung, die Markdownbefehle unterstützt, und listet die zu dem jeweiligen Stack gehörenden Messungen, unterteilt in Zeitreihen und Ortskur-

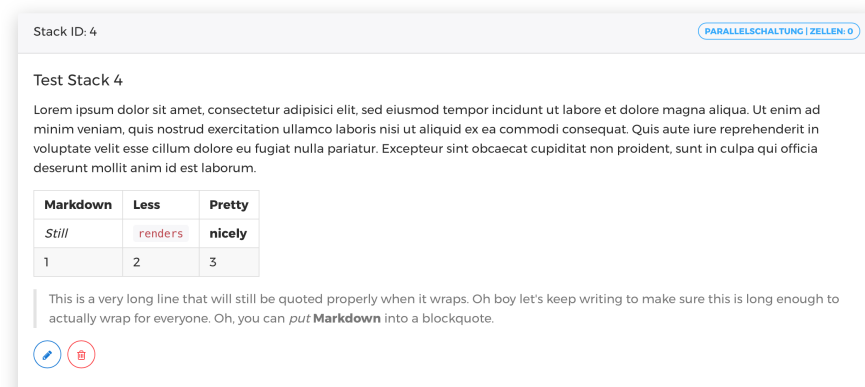


Abbildung 5: Stack-Detail-Ansicht

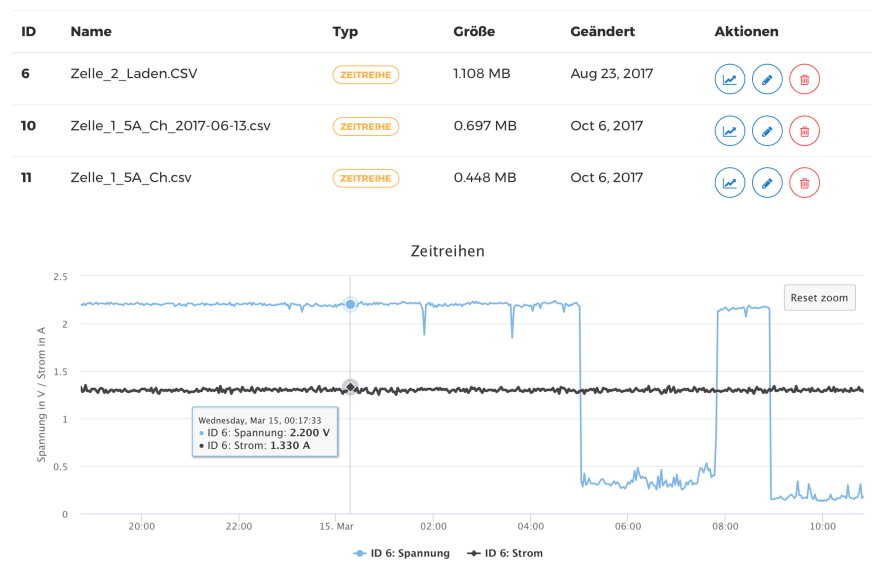
Zeitreihen <sup>3</sup>

Abbildung 6: Stackmessung

ven, in einer Tabellenansicht ähnlich der Ansicht der Messungen in der Datenbank, auf.

Klickt man nun auf eine Messung wird sie unter der entsprechenden Tabelle geplottet, wie in Abbildung 6 dargestellt.

Der Klick auf eine der Messungen führt in die Detailansicht der jeweiligen Messung. Dort werden neben Details zur Messung auch jedes Mal die Messung als Plot dargestellt. Zeitreihen sind wie bei allen anderen Plots zoombar, wie in Abbildung 7 (vorher) und Abbildung 8 (nachher) dargestellt. Die Messdaten werden automatisch vom Server nachgeladen um den Plot mit einer höheren Auflösung darzustellen.

Bei Zeitreihen mit unterschiedlichen Größen können einzelne Zeitreihen durch Klick auf die Legende ein- und ausgeblendet werden um sie so vergrößert darzustellen, wie in Abbildung 9 (Ladung aktiviert)

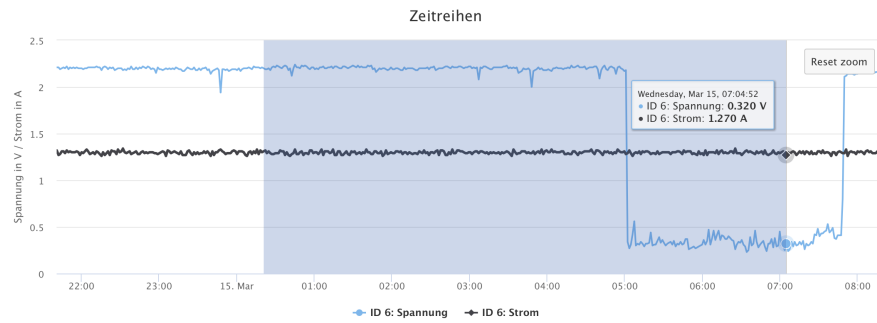


Abbildung 7: Zoom-Funktion vorher

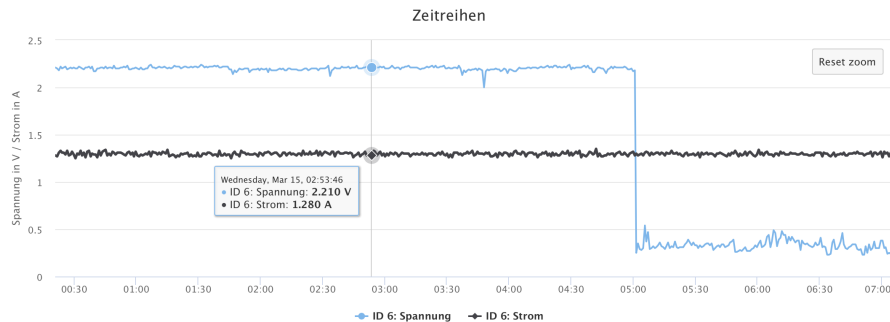


Abbildung 8: Zoom-Funktion nachher

10 (Ladung deaktiviert) dargestellt. Die Skalierung erfolgt automatisch.

Ein Plot einer Impedanzmessung ist in Abbildung 11 dargestellt. Die Koordinaten in der komplexen Ebene, wie auch die Frequenz sind am Cursor ablesbar.

Bearbeitet oder löscht man Datensätze, oder fügt man neue durch Hochladen hinzu, wird das Ergebnis in Form von Notifications bzw. Fehlermeldungen im oberen rechten Bildschirmrand angezeigt. Dies ist in Abbildung 12 einmal für beide Fälle dargestellt.

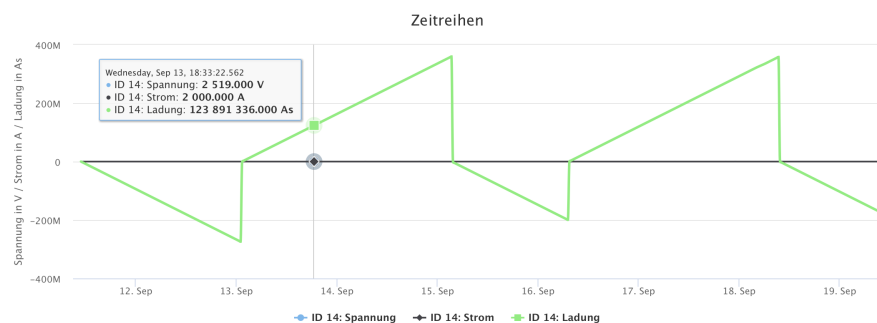


Abbildung 9: Ladungsplot aktiviert nachher

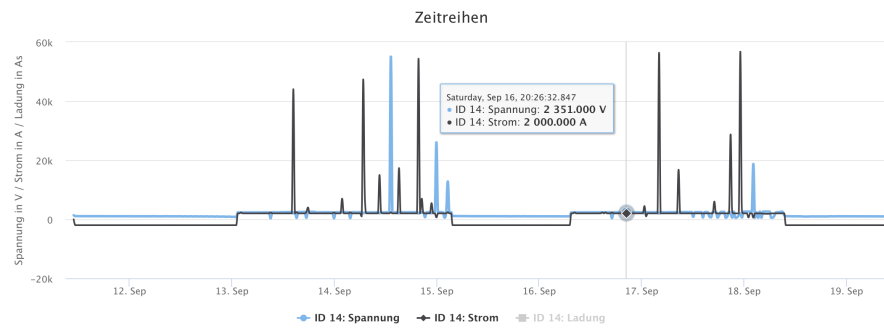


Abbildung 10: Ladungsplot deaktiviert

## Testmessung Ortskurve

Dies ist eine Beschreibung.

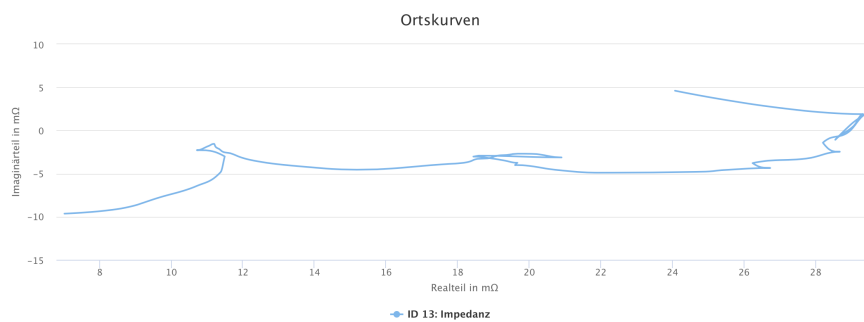
GRÖSSE:  
19.857 MBDATEINAME:  
Ortskurve.jsonGEMESSEN AM:  
Apr 1, 2017, 11:45 PMGEHÖRT ZU:  
Zelle 4

Abbildung 11: Detailansicht einer Ortskurve

**Löschen erfolgreich!**

Test Messung 1 wurde erfolgreich gelöscht!

**Error!**

Impedanz50.json konnte nicht gelöscht werden!

**ert**

**Aktionen**

2017

Abbildung 12: Notifications

## ZUSAMMENFASSUNG

---

### 3.1 FAZIT

### 3.2 AUSBLICK



## LITERATUR

---

- [1] ASP.NET. *Kestrel GitHub Repo*. [Accessed 01 October, 2017]. URL: <https://github.com/aspnet/KestrelHttpServer>.
- [2] Creative Tim Design Agency. *Now UI Kit*. [Accessed 07 October, 2017]. URL: <https://www.creative-tim.com/product/now-ui-kit>.
- [3] Google Inc. *Angular Frontend-Framework*. [Accessed 5 October, 2017]. URL: <https://angular.io>.
- [4] Google Inc. *Angular GitHub Repo*. [Accessed 5 October, 2017]. URL: <https://github.com/angular>.
- [5] Highsoft. *High Charts*. [Accessed 06 October, 2017]. URL: <https://www.highcharts.com>.
- [6] Hwaci. *SQLite*. [Accessed 29 May, 2017]. URL: <https://www.sqlite.org>.
- [7] Microsoft and the .NET Community. *Home repository for .NET Core*. [Accessed 28 April, 2017]. URL: <https://github.com/dotnet/core>.
- [8] Microsoft. *Entity Framework Core*. [Accessed 29 May, 2017]. URL: <https://docs.microsoft.com/en-us/ef/core/index>.
- [9] Twitter Inc. *Bootstrap CSS Framework*. [Accessed 07 October, 2017]. URL: <http://getbootstrap.com>.