# SWO5 - UE03 Dokumentation

Aufwand: 12h

## Inhaltsverzeichnis

1.	Architekturbeschreibung:	3
	1.1 Erweiterung UE03	4
	1.2 Domäne	5
	1.2.1 Ergänzungen Übung 3	5
	1.3. Logic	6
	1.4 Persönliche Änderungen	6
2.	Quellcode	6
3.	Unit-Tests	7
3.	Inbetriebnahme	8
4.	Benutzerdokumentation - WPF	9
	4.1. Übersicht	9
	4.2. Filter	.10
	4.3. Diagramm-Ansicht	.12
	4.4. Tabellen-Ansicht	. 13
	4.5. Export-Funktion	. 14
5.	Benutzerdokumentation – Web / ASP.NET	.16
	5.2 Bundesländer-Verwaltung	.16
	5.1.1 Erstellen eines neuen Bundeslandes	. 17
	5.1.2 Editieren eines bestehenden Bundeslandes	.18
	5.1.3 Entfernen eines bestehenden Bundeslandes	. 19
	5.2 Bezirke	20
	5.2.1 Erstellen eines neuen Bezirks	. 21
	5.2.2 Editieren eines bestehenden Bezirks	. 22
	5.2.3 Entfernen eines bestehenden Bezirks	23
	5.3 Benutzer	. 24
	5.3.1 Erstellen von neuen Benutzern	. 25
	5.3.2 Editieren eines bestehenden Benutzers	. 27
	5.3.3 Entfernen eines bestehenden Benutzers	. 29
6.	REST API	30
	6.1 Test der Methoden ohne Authorisierung	.30
	6.2 Test der Methoden mit Authorisierung	. 30

Abbildung 1: Klassendiagramm	
Abbildung 2: UML Übung 03	4
Abbildung 3: ER-Diagramm	5
Abbildung 4: Unit Tests	
Abbildung 5: How-To Datenbank hinzufügen	
Abbildung 6: How-to: Daten zur Datenbank hinzufügen	8
Abbildung 7: UI - Übersichts-Tab	9
Abbildung 8: UI - Bundesland Dropdown	. 10
Abbildung 9: UI - Bezirk Dropdown	. 10
Abbildung 10: UI - Zeitraum Dropdown	. 11
Abbildung 11: UI Datumsauswahl	. 11
Abbildung 12: UI - Diagrammansicht - Gesamt	. 12
Abbildung 13: UI - Diagrammansicht - Infiziert	. 13
Abbildung 14: UI – Tabellenansicht	. 13
Abbildung 15: Export Dialog	. 14
Abbildung 16: Export File	. 14
Abbildung 17: Export Ergebnis	. 15
Abbildung 18: Start – Bundesland – Anzeige	
Abbildung 19: Menüauswahl im Header	. 16
Abbildung 20: Bundesland Erstellung - Required	. 17
Abbildung 21: Bundesland Erstellung – Mindestlänge	
Abbildung 22: Bundesland Erstellung – Erfolgreich	. 17
Abbildung 23: Bundesland Editierung - Required	. 18
Abbildung 24: Bundesland Editierung - Mindestlänge	. 18
Abbildung 25: Bundesland Editierung – Erfolgreich	. 18
Abbildung 26: Bundesland Entfernung - Bezirk-Referenzen	. 19
Abbildung 27: Bundesland Entfernung - Keine Referenz	. 19
Abbildung 28: Bundesland Entfernung – Erfolgreich	. 19
Abbildung 29: Bezirke – Anzeige	. 20
Abbildung 30: Bezirk Erstellung - Required	
Abbildung 31: Bezirk Erstellung – Mindestlänge	
Abbildung 32: Bezirk Erstellung – Erfolgreich	
Abbildung 33: Bundesland Entfernung - Referenzierte Eilmeldungen	
Abbildung 34: Bundesland Entfernung - Keine Referenzen	. 23
Abbildung 35: Bundesland Entfernung – Erfolgreich	
Abbildung 36: Benutzer - Anzeige	
Abbildung 37: Benutzer Erstellung – Required	
Abbildung 38: Benutzer Erstellung - Mindestlänge	
Abbildung 39: Benutzer Erstellung - Passwort bestätigen	
Abbildung 40: Benutzer Erstellung - Erfolgreich	
Abbildung 41: Benutzer Editierung - Altes Passwort	
Abbildung 42: Benutzer Editierung - Nur Name	
Abbildung 43: Benutzer Editierung - Neues Passwort	
Abbildung 44: Benutzer Entfernung – Referenzierte Eilmeldungen	
Abbildung 45: Benutzer Entfernung - Keine Referenzen	
Abbildung 46: Benutzer Entfernung – Erfolgreich	
Abbildung 47: API Abfrage - Keine Authorisierung	
Abbildung 48: API Login – Token	
Abbildung 49: API Abfrage - Alle Reports	
Abbildung 50: API Abfrage - Ein Report	
Abbildung 51: API POST - Erstellung	. 33

## 1. Architekturbeschreibung:

Das Projekt orientiert sich stark an demjenigen Projekt, welches im Laufe der Übung durchbesprochen wurde, weshalb der Großteil der Struktur gleich ist. Das entworfene Klassendiagramm sieht wie folgt aus:

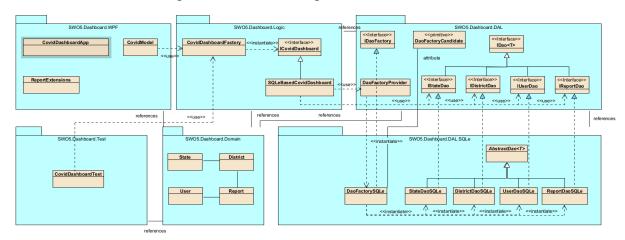


Abbildung 1: Klassendiagramm

Die zwei größten Änderungen stellen die Komponenten "Domain" und "Logic" dar, was auch Sinn macht, da das grundlegende Gerüst gleich ist, während sich aber die Daten und die Anwendungsfälle ändern.

Zudem habe ich mich dazu entschieden die Verbindung des DAL-Package mit der Implementierung wie in der Übung zu gestalten, weil ich das Konzept von Attributes und Assemblies verinnerlichen wollte. Ich kann mir vorstellen kann, dass derartige Flexibilität, auch wenn in diesem Fall nicht gebraucht, sehr hilfreich sein kann.

Dadurch, dass sich Domäne und Logik am meisten verändert haben, möchte ich in den nächsten Kapiteln noch näher auf meine Entscheidungen eingehen.

### 1.1 Erweiterung UE03

Als Erstes möchte ich kurz darauf hinweisen, dass ich leider auf die Web-Version von Visual-Paradigm ausweichen musste, und deswegen zum einen leichte Inkonsistenzen in der Datendarstellung habe, sowie nur den neuen Teil des UMLs samt deren Abhängigkeiten darstelle.

Für die Erstellung der Webanwendung wurden Controller für jeden Datentyp, sowie die zugehörigen Razor-Views und Viewmodels für das Model-Binding im Projekt "SWO5.Dashboard.Web" implementiert.

Zusätzlich dazu, wurde das Logic Package überarbeitet, sodass nun IManager existieren, die die bestehenden Daos verwenden, um einen einfachen Datenzugriff zu ermöglichen, ohne die gesamte Businesslogik (bzw. das vorherige CovidDashboard) einzubinden.

Für die Erstellung der API wurden entsprechende DTOs erstellt, welche den Klassen aus der Domain gleichkommen. Zusätzlich dazu wurden die notwendigen Controller für die Verwendung der Report-Requests sowie für den Login erstellt.

Überarbeitet wurde also das Logic-Projekt, während das Web, das DTO und das API Projekt hinzugekommen sind.

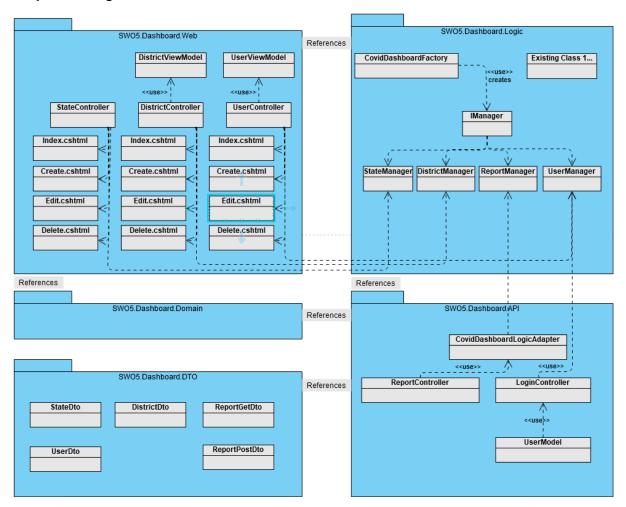


Abbildung 2: UML Übung 03

### 1.2 Domäne

Um die Entscheidungen besser erklären zu können, möchte ich kurz das Datenbankschema illustrieren, da die Klassen in C# ähnlich aufgebaut sind:

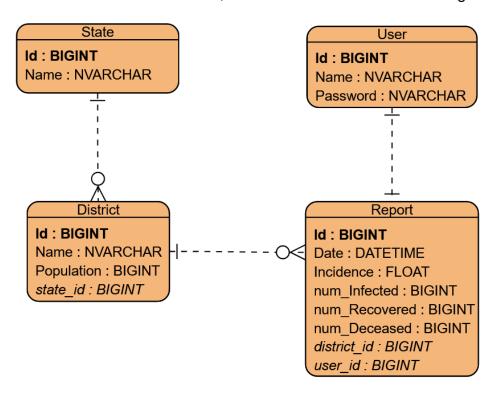


Abbildung 3: ER-Diagramm

Auf den ersten Blick wurden alle relevanten Daten in den Tabellen/Klassen gekapselt.

Anfangs dachte ich daran für District einen Composite-Key zu verwenden, was sich aber mit der Lösung aus dem Unterricht nicht gut vereinbaren lasst, da es wünschenswert ist, dass beim Einfügen bzw. Updaten die ID des Elements retourniert wird um dieses lokal zu vervollständigen bzw. eine Rückmeldung zu erhalten, ob die Datenbankbefehle erfolgreich waren oder nicht. Deswegen habe ich schlussendlich die gesamte GKZ (die indirekt das Bundesland enthält) als Key definiert.

Im Zuge meines Designs habe ich mich außerdem dazu entschieden nicht nur die IDs der Fremd-Entitäten in der C# Implementierung zu speichern, sondern vollständige Referenzen auf die entsprechenden Instanzen, um zu jeder Zeit vollständige Informationen zu haben.

### 1.2.1 Ergänzungen Übung 3

Im Laufe der Übung 3 mussten auch neue Bezirke/Bundesländer erstellt werden, was sich mit meiner bestehenden Datenbank schwierig gestaltete, da meine Primary Keys für diese Tabellen keine IDENTITY Spalten waren und somit die Keys nicht auto-inkrementierten. Das, sowie einige kleine damit verbundene Änderungen im DAL, wurden im Laufe dieser Übung eingearbeitet.

### 1.3. Logic

Wie im Unterricht wurde eine entsprechende Factory und ein entsprechendes Interface für alle gekapselten Funktionalitäten erstellt.

Für die User-Verwaltung wurden die Methoden UserExists und HashPassword angefügt, da ich die Passwörter sicher speichern will und ich mir vorstellen kann, dass es relevant ist, ob ein User (mit bestimmtem Namen) bereits existiert.

Außerdem wurde eine Funktion vorgesehen, welche ein gegebenes CSV der AGES ausliest und dessen Daten in die Datenbank schreibt.

Im Vergleich zur letzten Iteration (Übung 1) wurden die Funktionen, welche ein gewisses Zeitfenster betreffen, entfernt, da diese Funktionalität auch einfach in den entsprechenden Funktionen im GUI verwirklicht werden kann, wo ohnehin LINQ benötigt wird.

Hinzugekommen sind jedoch die Unit-Tests für alle Datenbank-Operationen, die Bundesländer und Bezirke betreffen. Bei der letzten Iteration wurden diese bewusst weggelassen, da diese Datensatze normalerweise konstant bleiben. Diese wurden vollständigkeitshalber ergänzt, da es bei anderen Personen deswegen zu einem Punkteabzug gekommen ist.

Wie bereits beim UML erwähnt sind bei der Logik im Rahmen der Übung 3 nur die Manager-Klassen hinzugekommen.

## 1.4 Persönliche Änderungen

Ich habe mich, konträr zur Übung, dazu entschieden, dass ich die JOINS wirklich in den Commands/Selects selbst verwirklichen will und nicht nachträglich im Code die Einträge zusammenstöpseln muss. Daraus ergibt sich, dass sämtliche Read-Methoden für District und Report auch in der spezialisierten Klasse genauer ausgeführt werden müssen, da sich die select-statements nicht mehr in AbstractDao zusammenfassen lassen.

Als Lösung dafür würde ich eine dynamische Bindung implementieren, indem ich die generell formulierten Methoden in AbstractDao als virtual kennzeichne und dann in den relevanten Subklassen überschreibe.

### 2. Quellcode

Der Quellcode wurde aufgrund der schieren Menge an Dateien nicht direkt ins PDF kopiert, weshalb ich hier auf das Visual Studio Projekt verweise.

Um den Code in Betrieb zu nehmen, muss vorher noch die Datenbank mithilfe des beigelegten SQL-Scripts erstellt und befüllt werden. Mehr dazu auch in der Benutzerdokumentation.

### 3. Unit-Tests

Ähnlich zum Quellcode werden auch diese im Detail im Code durchgegangen.

Die Testfälle, welche alle Funktionen der Business-Logik abdecken, waren leider teils (nicht vermeidbar) abhängig voneinander, es wurde aber so gut wie möglich versucht, dies zu reduzieren.

Anbei noch ein Screenshot des erfolgreichen Durchlaufs:

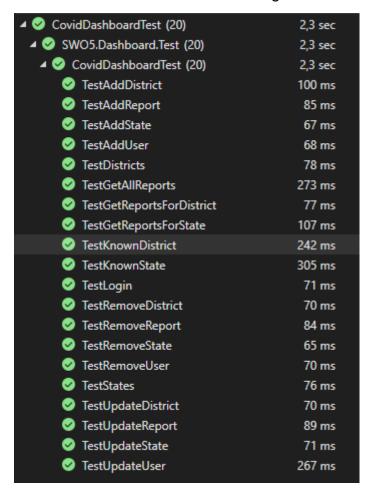


Abbildung 4: Unit Tests

Wie oben bereits erwähnt, wurden die Funktionen der Business-Logik um die CRUD-Methoden der Bundesländer und Bezirke erweitert bzw. die Funktionen mit Zeitfenstern entfernt.

### 3. Inbetriebnahme

Um das Programm in Betrieb nehmen zu können, muss zuerst die Datenbasis importiert werden. Die gesamte Datenstruktur, sowie alle Meldungen und zusätzlich notwendigen Daten können mithilfe der Datei "CREATE\_DATABASE.sql" angelegt werden.

Hierfür wird in Visual Studio im SQL Server Object Explorer eine Query auf den zuständigen SQL-Express-Server angelegt und eine Datenbank namens "Covid" erstellt (siehe erste Zeile im SQL-File).

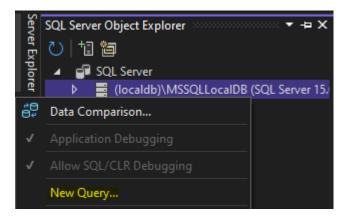


Abbildung 5: How-To Datenbank hinzufügen

Danach kann wiederum eine Verbindung/Query zu der neu erstellten Datenbank aufgebaut werden, um mit den restlichen Zeilen alle Daten anzulegen.

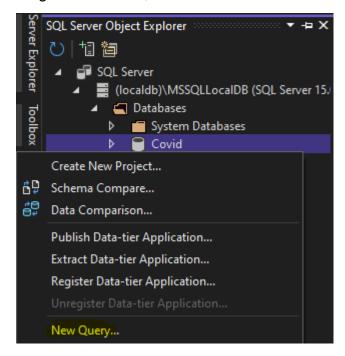


Abbildung 6: How-to: Daten zur Datenbank hinzufügen

### 4. Benutzerdokumentation - WPF

In diesem Bereich werden die wichtigsten Funktionen der Desktopanwendung erläutert.

### 4.1. Übersicht

Beim Programmstart wird der Benutzer vom UI im "Übersicht"-Tab begrüßt. In diesem sind die momentan aktiven Fälle, die neuen Fälle für den heutigen Tag, sowie eine Übersicht über die aktuellen 7-Tage-Inzidenzen aller Bundesländer zu sehen. Diese Felder sind von den Einstellungen oben unbeeinflusst und werden nur zum Start des Programms berechnet.

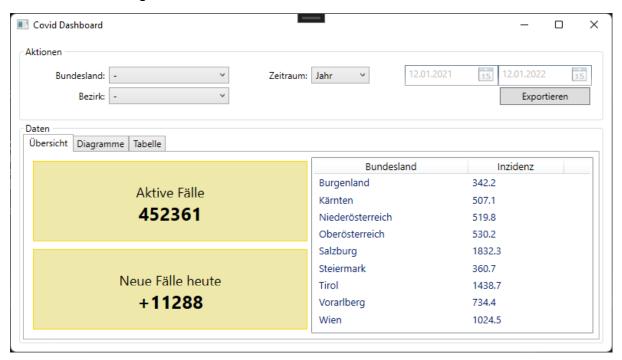


Abbildung 7: UI - Übersichts-Tab

### Anmerkung für Tutoren:

(Stand 10.12.2022, neuester Datensatz)

Die momentanen aktiven Fälle fallen bei meiner Berechnung deutlich höher aus als erwartet (ist: 452361, soll: 77165). Im Endeffekt sollten sich die aktiven Fälle aus der Differenz zwischen Summe der Infizierten Personen und der Summe der Genesenen/Verstorbenen Personen ergeben. Also praktisch Insgesamt\_Infiziert – (Insgesamt\_Genesen + Insgesamt\_Verstorben) – das scheint jedoch bei diesem Datenset nicht zu funktionieren.

Die neuen Fälle weichen leicht ab mit einem erwarteten Ergebnis von 10842 im Vergleich zu dem hier präsentierten Ergebnis von 11288.

### 4.2. Filter

Oben in der Ansicht zu sehen, sind die Filter welche festgelegt werden können. Über Dropdowns kann eine Kombination von Bundesland, Bezirk und einem Datumsbereich ausgewählt werden, anhand dessen die Diagramme und die Tabelle neu berechnet werden.

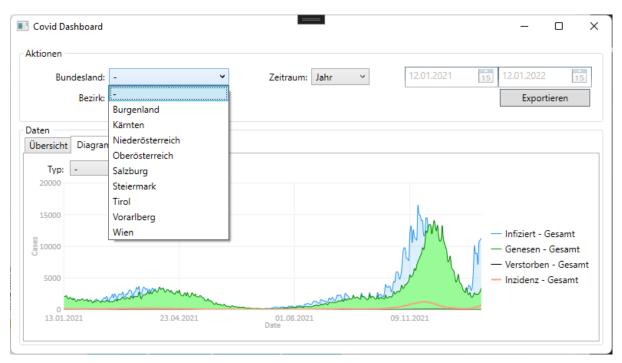


Abbildung 8: UI - Bundesland Dropdown

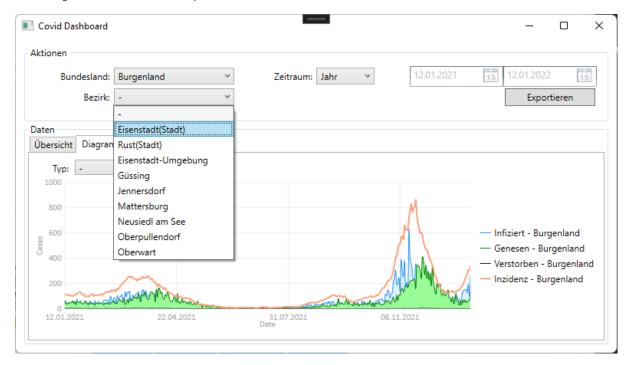


Abbildung 9: UI - Bezirk Dropdown

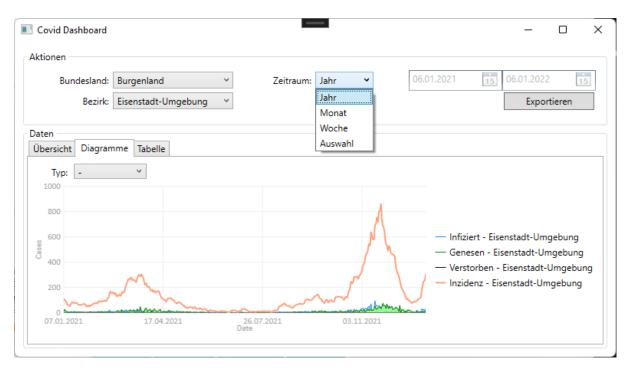


Abbildung 10: UI - Zeitraum Dropdown

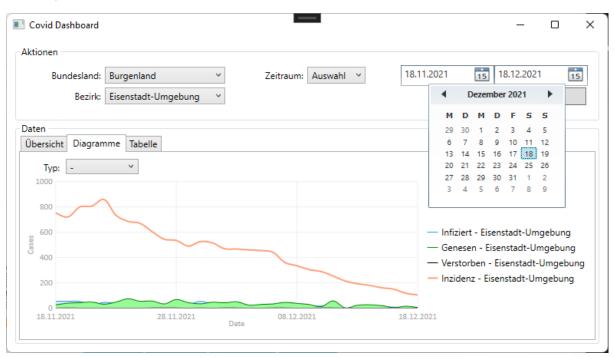


Abbildung 11: UI Datumsauswahl

### 4.3. Diagramm-Ansicht

Klickt man auf den Tab "Diagramme", welcher schon während der Präsentation der Filter hergezeigt wurde, so erhält man einen Überblick über das Infektionsgeschehen in Österreich bzw. in den ausgewählten Regionen im gewählten Zeitraum.

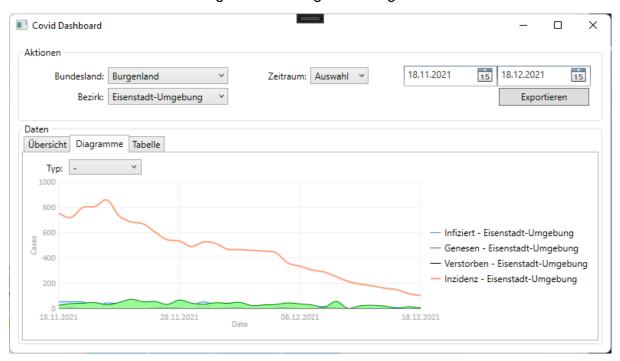


Abbildung 12: UI - Diagrammansicht - Gesamt

Standardmäßig werden alle Graphen gleichzeitig angezeigt, jedoch führt das zum Beispiel in diesem Szenario dazu, dass die Werte der Infektionen, Genesungen und Tode untergeht, da die Inzidenz zu hoch ist.

Aus diesem Grund, ist es möglich über die Combobox "Typ" auszuwählen, welcher Graph dargestellt werden soll. Die Werte werden auch entsprechend für eine anständige Darstellung skaliert:

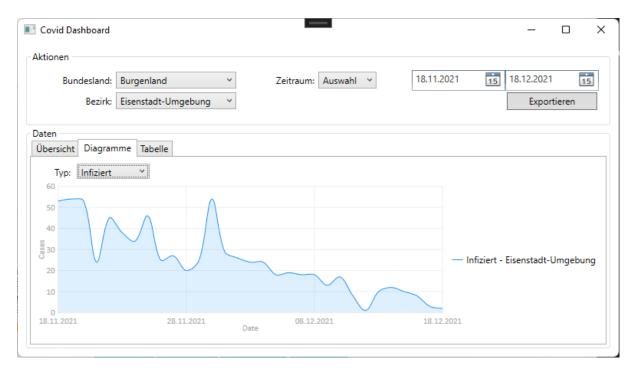


Abbildung 13: UI - Diagrammansicht - Infiziert

### 4.4. Tabellen-Ansicht

Alternativ zum Diagramm können die Daten auch als Tabelle visualisiert werden.

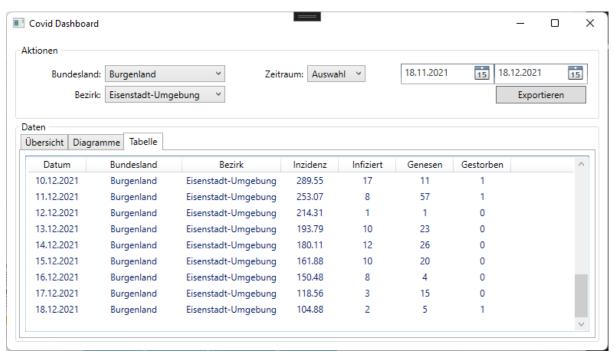


Abbildung 14: UI - Tabellenansicht

### 4.5. Export-Funktion

Über den Button "Export" können Benutzer die dargestellte Tabelle als CSV exportieren. Aufgrund der Tatsache, dass der Decimal-Point im deutschen das Komma ist, wurde als Delimiter das Semicolon verwendet.

Für einen Export muss der Nutzer nur den Button "Exportieren" anklicken und einen Speicherort festlegen.

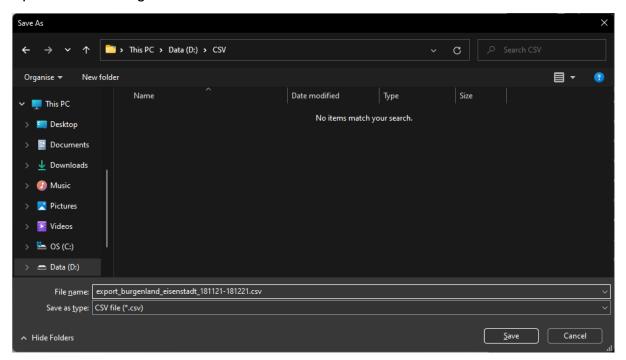


Abbildung 15: Export Dialog

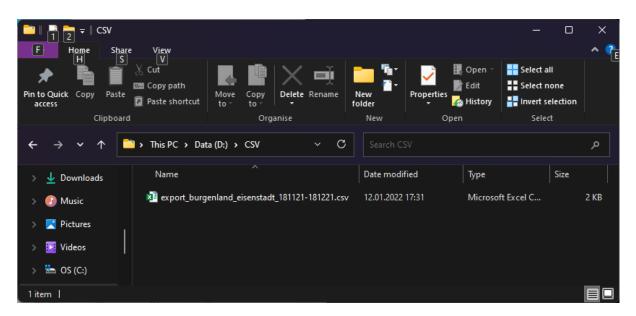


Abbildung 16: Export File

```
🔚 export_burgenland_eisenstadt_181121-181221.csv 🔀
     18.11.2021; Burgenland; Eisenstadt-Umgebung; 53; 26; 0; 754, 6567
     19.11.2021; Burgenland; Eisenstadt-Umgebung; 54; 39; 1; 720, 4578
     20.11.2021;Burgenland;Eisenstadt-Umgebung;53;43;0;800,2554
     21.11.2021;Burgenland;Eisenstadt-Umgebung;24;48;0;807,0952
  4
     22.11.2021;Burgenland;Eisenstadt-Umgebung;45;31;0;859,5335
     23.11.2021;Burgenland;Eisenstadt-Umgebung;38;48;0;734,1374
     24.11.2021; Burgenland; Eisenstadt-Umgebung; 34; 74; 0; 686, 2589
     25.11.2021;Burgenland;Eisenstadt-Umgebung;46;54;0;670,2994
  8
     26.11.2021;Burgenland;Eisenstadt-Umgebung;25;56;1;604,1814
  9
     27.11.2021; Burgenland; Eisenstadt-Umgebung; 27; 33; 1; 544, 9032
 10
     28.11.2021;Burgenland;Eisenstadt-Umgebung;20;68;1;535,7835
 11
     29.11.2021;Burgenland;Eisenstadt-Umgebung;25;43;0;490,1849
 12
     30.11.2021;Burgenland;Eisenstadt-Umgebung;54;34;1;526,6638
     01.12.2021;Burgenland;Eisenstadt-Umgebung;29;47;0;515,2642
     02.12.2021;Burgenland;Eisenstadt-Umgebung;26;42;1;469,6655
 15
     03.12.2021;Burgenland;Eisenstadt-Umgebung;24;50;1;467,3856
 16
 17
     04.12.2021;Burgenland;Eisenstadt-Umgebung;24;24;0;460,5458
 18
     05.12.2021; Burgenland; Eisenstadt-Umgebung; 18; 29; 0; 455, 986
     06.12.2021; Burgenland; Eisenstadt-Umgebung; 19; 33; 1; 442, 3064
 19
     07.12.2021;Burgenland;Eisenstadt-Umgebung;18;45;1;360,2289
 20
     08.12.2021;Burgenland;Eisenstadt-Umgebung;18;37;0;335,1497
 21
 22
     09.12.2021;Burgenland;Eisenstadt-Umgebung;13;29;0;305,5106
     10.12.2021;Burgenland;Eisenstadt-Umgebung;17;11;1;289,5511
 23
 24
     11.12.2021;Burgenland;Eisenstadt-Umgebung;8;57;1;253,0722
     12.12.2021;Burgenland;Eisenstadt-Umgebung;1;1;0;214,3134
     13.12.2021;Burgenland;Eisenstadt-Umgebung;10;23;0;193,794
 26
     14.12.2021; Burgenland; Eisenstadt-Umgebung; 12; 26; 0; 180, 1145
     15.12.2021; Burgenland; Eisenstadt-Umgebung; 10; 20; 0; 161, 875
 28
     16.12.2021;Burgenland;Eisenstadt-Umgebung;8;4;0;150,4754
 29
     17.12.2021;Burgenland;Eisenstadt-Umgebung;3;15;0;118,5564
 30
     18.12.2021; Burgenland; Eisenstadt-Umgebung; 2; 5; 1; 104, 8768
```

Abbildung 17: Export Ergebnis

## 5. Benutzerdokumentation – Web / ASP.NET

In diesem Teil wird die Funktion der Web-Implementierung zur Verwaltung der Stammdaten erläutert.

### 5.2 Bundesländer-Verwaltung

Beim Aufrufen der Anwendung/Website wird der User durch die Anzeige der Bundesländer begrüßt.

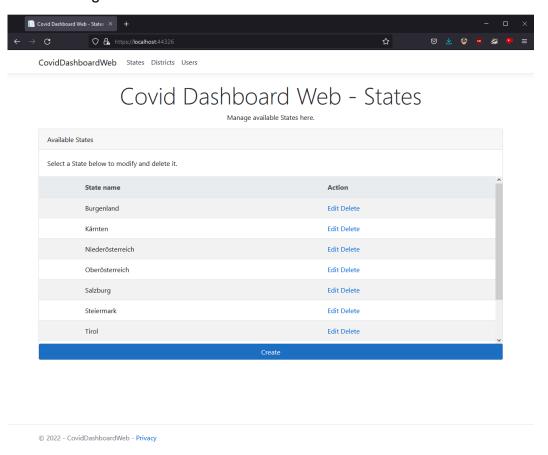


Abbildung 18: Start - Bundesland - Anzeige

In diesem Menü werden nicht nur die Daten jedes Bundeslandes angezeigt – der User kann zudem neue Bundesländer erstellen oder bestehende Bundesländer editieren und löschen.

Um in ein anderes Menü zu wechseln, können die Links im Header-Bereich verwendet werden:

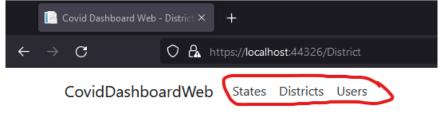


Abbildung 19: Menüauswahl im Header

### 5.1.1 Erstellen eines neuen Bundeslandes

Beim Erstellen des Bundeslandes muss der Benutzer lediglich den Namen des Bundeslandes eingeben. Dieser muss zwischen 2 und 50 Zeichen lang sein und ist Voraussetzung für die korrekte Erstellung. Werden diese Bedingungen nicht erfüllt, kommt es zur Fehlermeldung:



Abbildung 20: Bundesland Erstellung - Required



Abbildung 21: Bundesland Erstellung - Mindestlänge

Werden die Bedingungen erfüllt, so findet man den Eintrag nach Umleitung auf die Hauptseite in der Liste:

## Covid Dashboard Web - States

Manage available States here. Available States Select a State below to modify and delete it. State name Action Oberösterreich Edit Delete Salzburg Edit Delete Steiermark Edit Delete Edit Delete Tirol Vorarlberg Edit Delete Skellige Edit Delete

Abbildung 22: Bundesland Erstellung - Erfolgreich

### 5.1.2 Editieren eines bestehenden Bundeslandes

Beim Editieren wird das gleiche Formular wie beim Erstellen aufgerufen, jedoch vor ausgefüllt mit den bestehenden Daten. Bei der Editierung gelten die gleichen Regeln wie bei der Erstellung.



Abbildung 23: Bundesland Editierung - Required



Abbildung 24: Bundesland Editierung - Mindestlänge

Werden die Daten widerum korrekt eingegeben, so findet man das veränderte Element auf:

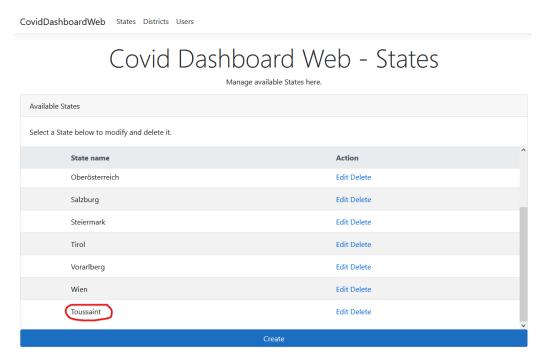
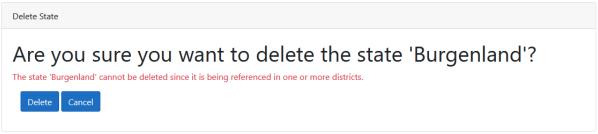


Abbildung 25: Bundesland Editierung - Erfolgreich

### 5.1.3 Entfernen eines bestehenden Bundeslandes

Beim Löschen von Bundesländern gibt es nicht viel zu beachten. Der User wird lediglich nochmals gefragt, ob er das Bundesland wirklich löschen will. Versucht der Benutzer jedoch ein Bundesland zu löschen, dass durch bestehende Bezirke referenziert wird, so erhält der Benutzer eine Fehlermeldung:



Back to List

Abbildung 26: Bundesland Entfernung - Bezirk-Referenzen

Wird ein Bundesland gelöscht, zu dem es keine Bezirke gibt, so funktioniert der Vorgang jedoch und das Element verschwindet aus der Liste.

Delete State
Are you sure you want to delete the state 'Toussaint'?
Back to List

Abbildung 27: Bundesland Entfernung - Keine Referenz

Available States		
Select a State below to modify and delete it.		
State name	Action	^
Niederösterreich	Edit Delete	
Oberösterreich	Edit Delete	
Salzburg	Edit Delete	
Steiermark	Edit Delete	
Tirol	Edit Delete	
Vorarlberg	Edit Delete	
Wien	Edit Delete	
<del></del>	Create	~

Abbildung 28: Bundesland Entfernung – Erfolgreich

### 5.2 Bezirke

Zusätzlich zu den Bundesländern können natürlich auch die Bezirke verwaltet werden. Ähnlich zu den Bundesländern können auch diese erstellt, editiert und entfernt werden.

Angezeigt wird anfangs allerdings wieder die Liste:

# Covid Dashboard Web - Districts

Manage available Districts here.

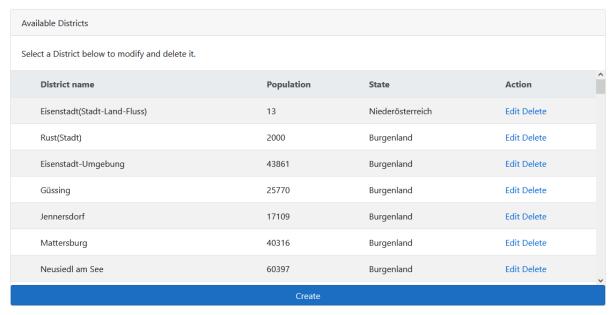


Abbildung 29: Bezirke - Anzeige

### 5.2.1 Erstellen eines neuen Bezirks

Beim Erstellen eines neuen Bezirks muss der Benutzer verpflichtend dessen Name, die Population, sowie das darüberliegende Bundesland auswählen.

Alle Felder sind verpflichtend. Zusätzlich gilt wieder die Einschränkung, dass der Name zwischen 2 und 50 Zeichen lang sein muss, sowie, dass die Population nur positive Zahlen annehmen darf.

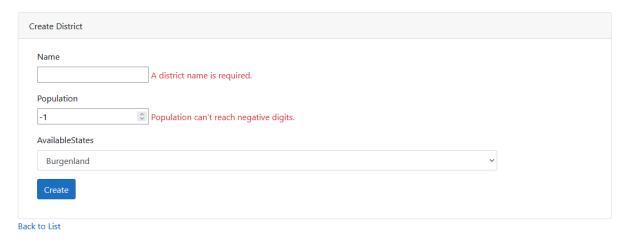


Abbildung 30: Bezirk Erstellung - Required

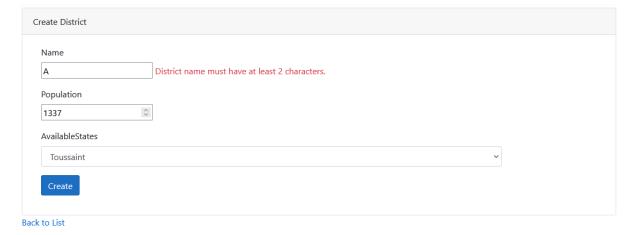


Abbildung 31: Bezirk Erstellung - Mindestlänge

Werden die Daten korrekt ausgefüllt, so findet sich der Bezirk in der Liste wieder:

# Covid Dashboard Web - Districts

Manage available Districts here.

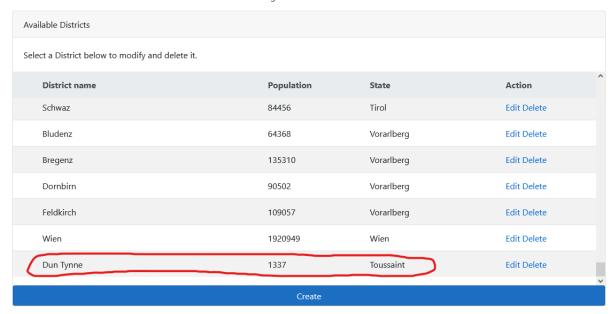
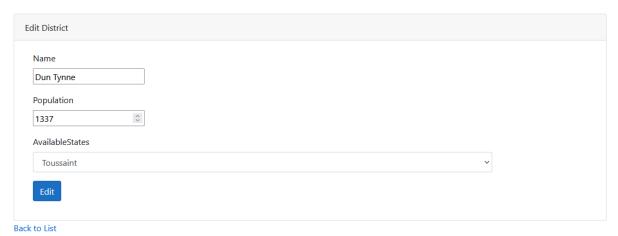


Abbildung 32: Bezirk Erstellung - Erfolgreich

### 5.2.2 Editieren eines bestehenden Bezirks

Wie bei den Bundesländern sieht das Editier-Formular wieder analog zum Erstell-Formular aus. Beim Aufrufen werden wieder die bestehenden Daten vorausgefüllt:



Auf die restlichen Anwendungsfälle bezüglich der Validierung bei der Editierung wird aufgrund der Redundanz nicht mehr eingegangen und kann im Zweifel durch Ausführen des Programms kontrolliert werden.

### 5.2.3 Entfernen eines bestehenden Bezirks

Wie beim Entfernen der Bundesländer wird auch hier zuerst der Nutzer nochmals gefragt, sowie beim Löschen überprüft, ob der Bezirk in einer Eilmeldung referenziert wurde.

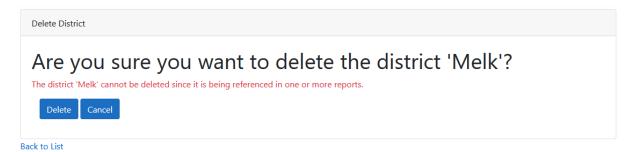


Abbildung 33: Bundesland Entfernung - Referenzierte Eilmeldungen



Back to List

Abbildung 34: Bundesland Entfernung - Keine Referenzen

# Covid Dashboard Web - Districts

Manage available Districts here.

Available Districts			
Select a District below to modify and delete it.			
District name	Population	State	Action
Reutte	33054	Tirol	Edit Delete
Schwaz	84456	Tirol	Edit Delete
Bludenz	64368	Vorarlberg	Edit Delete
Bregenz	135310	Vorarlberg	Edit Delete
Dornbirn	90502	Vorarlberg	Edit Delete
Feldkirch	109057	Vorarlberg	Edit Delete
Wien	1920949	Wien	Edit Delete
Y	Create		×

Abbildung 35: Bundesland Entfernung - Erfolgreich

### 5.3 Benutzer

Zu guter letzt können auch noch Benutzer in der Webanwendung verwaltet werden. Wiederum gestaltet sich dies analog zu Bezirken und Bundesländern.

# Covid Dashboard Web - Users

Manage available Users here.

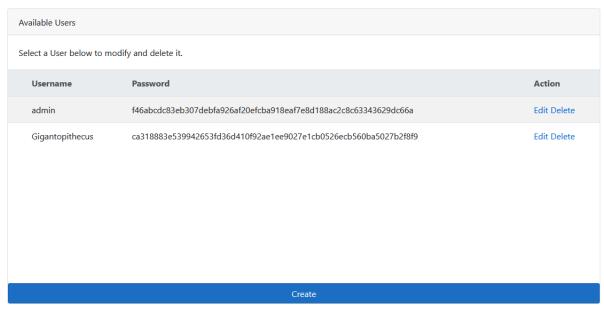


Abbildung 36: Benutzer - Anzeige

### 5.3.1 Erstellen von neuen Benutzern

Benutzer bestehen aus einem Namen und einem Passwort.

Der Name muss zwischen 2 und 50 Zeichen lang sein, während das Passwort mindestens aus 6 Zeichen bestehen muss und keine Maximal-Länge hat, da es ohnehin mittels SHA256 in einen 64-Zeichen langen Hash transformiert wird.

reate User		
Name	A username is required.	
Password		
A password is required.		
Confirm Password		
Create		
to List		

Abbildung 37: Benutzer Erstellung – Required

Create User		
Name C	Username must have at least 2 characters.	
Password		
Password must have at least 6	sharedore	
Confirm Password	cnaracters.	
Create		
ack to List		

Abbildung 38: Benutzer Erstellung - Mindestlänge

Hat der Benutzer valide Daten eingegeben, muss der Benutzer das Passwort zudem noch bestätigen, indem er es ein zweites Mal eintippt.

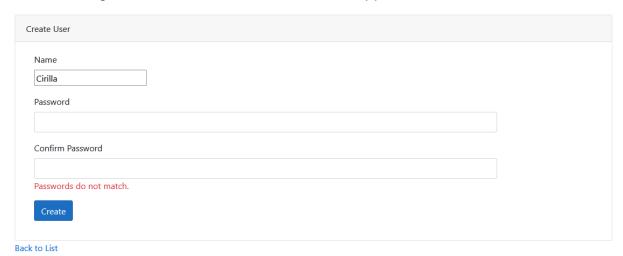


Abbildung 39: Benutzer Erstellung - Passwort bestätigen

Wurden die Daten richtig eingegeben und das Passwort bestätigt, so findet man den Benutzer mitsamt seinem Hash in der Liste wieder.



Abbildung 40: Benutzer Erstellung - Erfolgreich

### 5.3.2 Editieren eines bestehenden Benutzers

Beim Editieren eines Benutzers ist es wichtig, dass dieser sein bestehendes Passwort weiß, sodass niemand anders seine Daten editieren kann.

Es kann der Name sowie das Passwort geändert werden. Bei Leerlassen des "New Password"-Feldes, wird das Passwort beim Alten belassen.

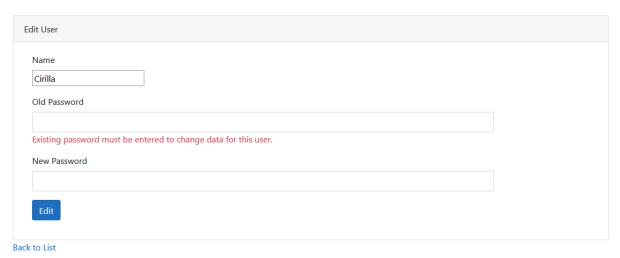


Abbildung 41: Benutzer Editierung - Altes Passwort

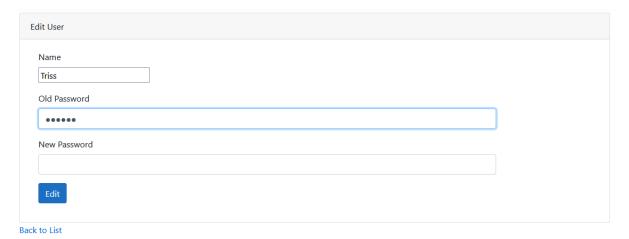




Abbildung 42: Benutzer Editierung - Nur Name

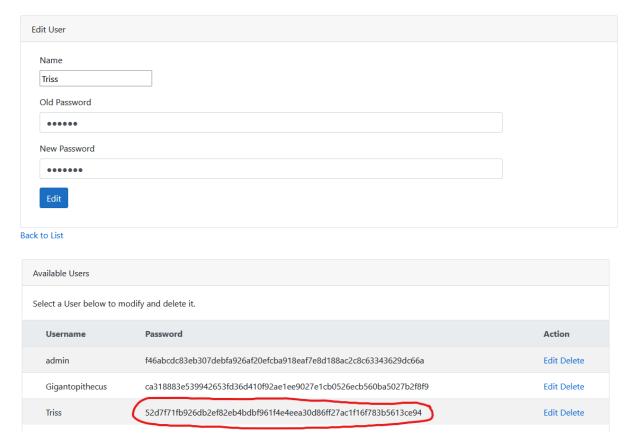


Abbildung 43: Benutzer Editierung - Neues Passwort

Es ist aufgrund der ähnlichen Hashes leider schwerer zu erkennen, jedoch wurde das Passwort geändert.

### 5.3.3 Entfernen eines bestehenden Benutzers

Für den Benutzer gilt beim Entfernen das gleiche wie beim Bezirk – er darf nicht in einem Report referenziert sein!

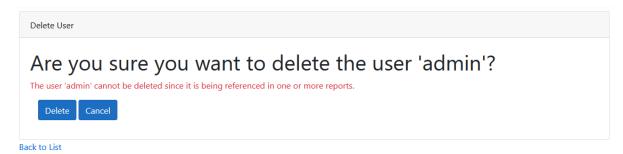
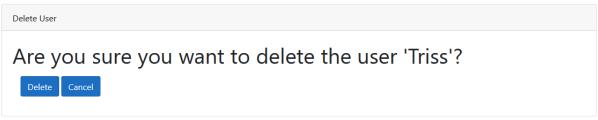


Abbildung 44: Benutzer Entfernung – Referenzierte Eilmeldungen



Back to List

Abbildung 45: Benutzer Entfernung - Keine Referenzen

Available Users	Available Users		
Select a User below to mo	odify and delete it.		
Username	Password	Action	
admin	f46abcdc83eb307debfa926af20efcba918eaf7e8d188ac2c8c63343629dc66a	Edit Delete	
Gigantopithecus	ca318883e539942653fd36d410f92ae1ee9027e1cb0526ecb560ba5027b2f8f9	Edit Delete	

Abbildung 46: Benutzer Entfernung - Erfolgreich

### 6. REST API

Für den Zugriff auf Eilmeldungen ist eine REST Schnittstelle vorgesehen. Mithilfe von dieser Schnittstelle können sich User authentifizieren und danach Reports abfragen und erstellen.

Für die Tests auf diese Schnittstelle bzw. Anleitungen zur Erstellung von gültigen Datenstrukturen in JSON liegt ein File "API\_TEST.http" vor, welches in VS Code geöffnet werden kann.

### 6.1 Test der Methoden ohne Authorisierung

Der Aufruf jeglicher Methoden ohne Authorisierung bzw. vorangehenden Login führt zu folgender Nachricht:

```
HTTP/1.1 401 Unauthorized
Transfer-Encoding: chunked
Server: Microsoft-IIS/10.0
WWW-Authenticate: Bearer
X-Powered-By: ASP.NET
Date: Sun, 06 Feb 2022 19:04:04 GMT
Connection: close
```

Abbildung 47: API Abfrage - Keine Authorisierung

### 6.2 Test der Methoden mit Authorisierung

Als Erstes muss sich der Benutzer anmelden. Das bereitgestellte File enthält einen fertigen User zur Anmeldung.

Stimmen die Daten mit jenen in der Datenbank überein, so wird ein Token retourniert:

```
HTTP/1.1 200 OK
Transfer-Encoding: chunked
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Sun, 06 Feb 2022 19:07:22 GMT
Connection: close

{
    "token": "eyJhbGci0iJIUzI1NiISInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1hcy54bWxzb2FwLm9
yZy93cy8yMDA1LzA1L2lkZW50aXR5L2NsYWltcy9uYW1laWRlbnRpZmllciI6ImFkbWluIiwiZXhwIjoxNjQ
0MTgxNjQyLCJpc3Mi0iJOdXRlbGxhLUVyZGJlZXJtYXJtZWxhZGUtV2FmZmVsLUdlbm9zc2VuY2x1YiIsImF
1ZCI6Ik51dGVsbGEtRXJkYmVlcm1hcm1lbGFkZS1XYWZmZWwtR2Vub3NzZW5jbHViIn0.TOUGXWunTDx8H87
ty1a8VIkoqoXzZfbojznZuQqSnsY"
}
```

Der oben erhaltene Token muss nach dem Login bei den Requests mit übergeben werden, um sich zu authorisieren. Der gegebene Token muss im File nur ausgetauscht werden, damit die Requests funktionieren.

Beim Abfragen "aller" Reports, musste die Größe eingeschränkt werden, da VS Code sowie der Browser sonst einfroren. Es werden daher nur 100 der ca. 65000 Datensätze retourniert.

```
HTTP/1.1 200 OK
Transfer-Encoding: chunked
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Sun, 06 Feb 2022 19:10:05 GMT
Connection: close
    "id": 64297,
    "date": "2022-01-10T00:00:00",
    "responsibleDistrict": {
      "id": 101,
      "responsibleState": {
        "id": 3,
        "name": "Niederösterreich"
      "name": "Eisenstadt(Stadt-Land-Fluss)",
      "population": 13
    },
    "infected": 12,
    "recovered": 10,
    "deceased": 0,
    "incidence": 422.9607,
    "fromUser": {
      "id": 0,
      "name": "admin",
      "password": "f46abcdc83eb307debfa926af20efcba918eaf7e8d188ac2c8c63343629dc6
6a"
    "id": 64298,
```

Abbildung 49: API Abfrage - Alle Reports

Alternativ dazu kann natürlich auch ein spezifischer Report (z.B. mit der ID 15000) abgefragt werden:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Sun, 06 Feb 2022 19:11:59 GMT
Connection: close
Content-Length: 334
  "id": 15000,
  "date": "2020-08-03T00:00:00",
  "responsibleDistrict": {
    "id": 411,
    "responsibleState": {
      "id": 4,
      "name": "Oberösterreich"
    "name": "Perg",
    "population": 69241
  },
  "infected": 3,
  "recovered": 1,
  "deceased": 0,
  "incidence": 14.44231,
  "fromUser": {
    "id": 0,
    "name": "admin",
    "password": "f46abcdc83eb307debfa926af20efcba918eaf7e8d188ac2c8c63343629dc66a"
  }
```

Abbildung 50: API Abfrage - Ein Report

Zu guter letzt kann auch noch ein Report erstellt werden:

```
HTTP/1.1 201 Created
Content-Type: application/json; charset=utf-8
Location: <a href="https://localhost:44330/api/CovidDashboard/64399">https://localhost:44330/api/CovidDashboard/64399</a>
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Sun, 06 Feb 2022 19:12:28 GMT
Connection: close
Content-Length: 273
{
  "id": 64399,
  "date": "2022-02-06T15:39:16.702Z",
  "responsibleDistrict": {
    "id": 103,
    "responsibleState": {
      "id": 0,
      "name": "string"
    },
    "name": "string",
    "population": 0
  },
  "infected": 420,
  "recovered": 1337,
  "deceased": 42,
  "incidence": 68.9,
  "fromUser": {
    "id": 0,
    "name": "string",
    "password": "string"
```