

# Abschlussprojekt in “Mobile Web Apps entwickeln”

## 2. Termin

27. April 2021

## 1 Abgabe

Erstellen Sie für die Ausarbeitung dieses Projekts einen neuen Ordner. Dieser Ordner soll alle Dateien enthalten, die Ihr Projekt benötigt. Schicken Sie *bis Sonntag, 2. Mai 2021 um 23:59 Uhr* eine Email an `alles.mil@gmail.com` mit

- dem Betreff “Mobile Web Apps Projekt”,
- Ihrem Namen als Nachricht und
- einem Downloadlink zu ihrem Projektordner als ZIP-Datei. Nachdem GMail Anhänge mit JavaScript-Dateien blockiert, muss der Projektordner auf einen Hoster wie zum Beispiel <https://wetransfer.com/> hochgeladen werden.

## 2 Projektbeschreibung

Erstellen Sie eine Web App für die Administration von Sendemasten in Österreich. Mitarbeiter einer Telekommunikationsfirma sollen damit einerseits eine Übersicht über die Sendemasten und die Netzabdeckung bekommen und andererseits sollen die Mitarbeiter die Daten von Sendemasten bearbeiten können. Erstellen Sie für diese beiden Teile jeweils eine eigene HTML-Datei.

Die Funktionalität der beiden Teile ist in Unterabschnitt 2.1 und Unterabschnitt 2.2 beschrieben. Auf die Daten der Sendemasten greifen Sie über eine Web API zu, die in Abschnitt 3 dokumentiert ist. Auch die Properties eines Sendemasten-Objects, wie es von der API geliefert wird, ist dort beschrieben. Die API wird Ihnen zur Verfügung gestellt und muss nicht von Ihnen entwickelt werden.

1. (5 Punkte) Achten Sie auf Lesbarkeit in Ihren HTML-, JavaScript- und CSS-Dateien indem Sie u.a. folgende Dinge beherzigen:
  - Sprechende Namen für Elemente, Klassen, Funktionen, Variablen
  - So wenige globale Variablen wie möglich
  - Saubere Einrückungen (Automatische Formatierung durch Ihre IDE!)
  - Semantisch sinnvolle HTML-Elemente
2. (5 Punkte) Behandeln Sie auch den Fehlerfall beim Holen von Daten aus dem Internet oder von Gerätesensoren. Geben Sie den Anwendern auch bei einem Fehler ein visuelles

Feedback. Erwarten Sie nicht, dass die Anwender die Developer Tools öffnen und die Console nach Fehlern durchsuchen.

## 2.1 Übersicht über Sendemasten

Auf dieser Seite wird die Übersicht über die Sendemasten dargestellt.

3. Diese Übersichtsseite soll folgende Funktionalität bieten:
  - (a) (5 Punkte) Die Sendemasten werden als Liste dargestellt. Zeigen Sie auf jeden Fall die Properties **name** und **id** von jedem Sendemasten an. Wählen Sie eine dritte Property und stellen auch diese dar. Beachten Sie, dass manche Properties eines Sendemasten auch HTML-Code enthalten können. Stellen Sie diese ausschließlich als Text dar.
  - (b) (5 Punkte) Jeder Eintrag in der Liste soll einen Link enthalten, um den jeweiligen Sendemasten auf der Bearbeitungs-Seite (siehe Unterabschnitt 2.2) zu bearbeiten.
  - (c) (5 Punkte) Zeigen Sie die Summe aller Sendemasten an, die 5G unterstützen (Property **is5GEnabled**). Wenn 3 oder weniger Sendemasten 5G unterstützen, soll diese Summe in rot und fett dargestellt werden.

## 2.2 Bearbeiten der Daten eines Sendemasten

4. Diese Seite zur Bearbeitung der Daten eines Sendemasten soll folgende Funktionalität bieten:
  - (a) (5 Punkte) Die Anwender erreichen diese Seite über den Link auf der Übersichtsseite (siehe Unterabschnitt 2.1). Die ID des Sendemasten wird dieser Seite als Search Parameter in der URL übergeben.
  - (b) (5 Punkte) Die Daten des Sendemasten sollen dargestellt werden. Verwenden Sie **input**- und **label**-Elemente.
  - (c) (5 Punkte) Bei kleinen Bildschirmbreiten sollen **label** und **input** jeweils in einer eigenen Zeile angezeigt werden. Bei größeren Breiten sollen sie nebeneinander angezeigt werden.
  - (d) (5 Punkte) Die Anwender sollen die Inputfelder für den Namen (Property **name**), die Reichweite (Property **range**) und die 5G-Unterstützung (Property **is5GEnabled**) bearbeiten können. Die **input**-Elemente der anderen Properties sollen nicht durch die Anwender verändert werden können.
  - (e) (5 Punkte) Nach dem Klick auf den Speichern-Button sollen die geänderten Daten für diesen Sendemasten zum Server geschickt werden. Die Anwender sollen das visuelle Feedback bekommen, ob das Speichern erfolgreich oder nicht war.

- (f) (5 Punkte) Darstellung des Sendemasten auf einer Karte. Verwenden Sie dazu Leaflet (<https://leafletjs.com/>). Stellen Sie die Position des Sendemasten als Marker und seine Reichweite als halbtransparent gefüllten Kreis auf der Karte dar. Als Kartenebene können Sie die OpenStreetMap verwenden.
- (g) (5 Punkte) Die Anwender sollen auch ihre eigene Position auf der Karte sehen, falls ihr Browser dies unterstützt und sie den Zugriff auf den Standort erlauben.

## 3 Web API

Die hier beschriebene Web API stellt Ihnen den Zugriff auf die Sendemasten der Telekomfirma zur Verfügung. Die API speichert die Daten nicht dauerhaft. Sollte ein Neustart des Servers notwendig sein, werden die Daten der Sendemasten wieder auf die Defaultwerte zurückgesetzt. Weiters arbeiten alle Prüflinge auf den selben Daten. Es kann also sein, dass Ihre Änderungen überschrieben werden, wenn jemand anders den gleichen Sendemasten bearbeitet.

### 3.1 Sendemasten-Object

Ein Sendemasten wird in der API als Object mit Properties für Identifikation, Name, Position, Reichweite und Kosten gespeichert. Im Folgenden werden die Properties beschrieben. Beachten Sie, dass nur **name**, **range** und **is5GEnabled** von den Anwendern Ihrer App verändert werden können sollen. Ein Beispiel für ein Sendemasten-Objekt ist in Listing 1 gegeben.

**id** Ein String, der den Sendemasten eindeutig identifiziert. Er kann aus Buchstaben und Zahlen bestehen. Diese Property kann von den Anwendern nicht bearbeitet werden.

**lat** Der Breitengrad der Position des Sendemasten als Dezimalzahl. Diese Property kann von den Anwendern nicht bearbeitet werden.

**lon** Der Längengrad der Position des Sendemasten als Dezimalzahl. Diese Property kann von den Anwendern nicht bearbeitet werden.

**cost** Die jährlichen Kosten für den Betrieb des Sendemasten in Euro als eine Dezimalzahl mit maximal zwei Nachkommastellen. Diese Property kann von den Anwendern nicht bearbeitet werden.

**name** Der Name des Sendemasten als String. Diese Property kann von den Anwendern bearbeitet werden.

**range** Eine positive Zahl die die Reichweite des Sendemasten in Metern angibt. Diese Property kann von den Anwendern bearbeitet werden.

**is5GEnabled** Eine boolesche Property die angibt, ob ein Sendemasten 5G unterstützt (**true**) oder nicht (**false**). Diese Property kann von den Anwendern bearbeiten werden.

```
1 {  
2   "id": "styria1",  
3   "lat": 47.394167,  
4   "lon": 13.689167,  
5   "cost": 18378.32  
6   "name": "Schladming",  
7   "range": 50,  
8   "is5GEnabled": false ,  
9 }
```

Listing 1: Beispiel eines Sendemasten-Objects

## 3.2 API Methoden

Die Web API ist erreichbar unter der Basis-URL <https://test.sunbeng.eu/api/> und bietet drei Endpunkte. Daten müssen als JSON zum Server geschickt werden. Der Server antwortet ebenfalls mit JSON. Für diese API ist Cross-Origin Resource Sharing (CORS) aktiviert.

### GET <https://test.sunbeng.eu/api/towers>

Gibt eine Liste von Sendemasten zurück. Der Server antwortet mit HTTP Status 200 und einer Liste von Sendemasten-Objects als JSON-String im Response-Body.

### GET [https://test.sunbeng.eu/api/towers/\\${id}](https://test.sunbeng.eu/api/towers/${id})

Gibt die Daten des Sendemasten mit der ID `id` zurück. Der Server antwortet mit dem HTTP Status 404, wenn kein Sendemasten mit der entsprechenden ID existiert. Ansonsten antwortet der Server mit dem HTTP Status 200 und dem Sendemasten-Object als JSON-String im Response-Body.

### POST [https://test.sunbeng.eu/api/towers/\\${id}](https://test.sunbeng.eu/api/towers/${id})

Speichert die Daten des Sendemasten mit der ID `id`. Der Body des HTTP-Request muss ein JSON-String mit den Properties für `name`, `range` und/oder `is5GEnabled` sein. Der Request muss nicht alle Properties beinhalten. Alle weiteren mitgeschickten Properties werden ignoriert. Wenn das Speichern erfolgreich war, antwortet der Server mit dem HTTP Status 200. Der Server antwortet mit dem HTTP Status 404, wenn kein Sendemasten mit der entsprechenden ID existiert. Der Server antwortet mit dem HTTP Status 400, wenn die Properties den falschen Datentypen haben oder `range` eine negative Zahl ist.

## 4 Punkte

Punkte	Note
0 - 30	5
31 - 37	4
38 - 45	3
46 - 53	2
54 - 60	1

Question:	1	2	3	4	Total
Points:	5	5	15	35	60
Score:					