Assignment 2 – TypeScript

Von Marco Scotellaro

Dozent: Josef Slezak

Studiengang: Onlinemedien

Studienjahr: 2020

Thema: Ich hab noch nie… Web-App

Inhalt

[1. Dokumentation 3](#_Toc39595271)

[1.1 Start HTML und CSS 3](#_Toc39595272)

[2. array.ts 3](#_Toc39595273)

[3. button.ts 3](#_Toc39595274)

[4. index.ts 4](#_Toc39595275)

[2. Reflexion 4](#_Toc39595276)

# 1. Dokumentation

## 1.1 Start HTML und CSS

Der Beginn meines Assignment 2 bestand für mich darin, meiner Lieblingsaufgabe nachzugehen. Ich wollte die Anwendungsoberfläche erstellen. Entschieden habe ich mich hierbei für ein schlichtes Design. Die Seite ist soweit Weiß gehalten und das Element besteht aus einem Div, einem <p>-Tag mit dem Inhalt „Ich hab noch nie…“, einem Default leeren <p>-Tag in welches die Aussage geschrieben wird, wenn man den Button betätigt. Das Ganze habe ich auch wieder in der Mitte des Bildschirmes platziert. Ich habe mich für eine serifenlose Schrift entschieden, da diese auf dem Bildschirm besonders gut zu lesen ist.

## 2. array.ts

Dann überlegte ich mir, wie ich am Besten die Logik der Web-App aufbauen möchte. Ich entschloss mich dazu, mit dem Array anzufangen. Ich hatte die Überlegung ein großes Array zu initialisieren. Dieses Array sollte alle Aussagesätze beinhalten, welche man „noch nie gemacht hat“. Ich habe für diese Web-App 49 Aussagesätze verwendet, welche ich von einer Website kopiert habe (https://psycatgames.com/de/magazine/party-games/never-ever/). Dieses Array ist eine Konstante und wird exportiert, da ich das Array in einem anderen Modul verwenden möchte. Das Array habe ich typisiert als ein Array aus Strings.

## 3. button.ts

Beginnen möchte ich mit der Dokumentation meines Codes. Anschließend stell ich dar, welche Probleme ich bezüglich des Codes hatte. Im Modul button.ts angekommen, habe ich zu Beginn das String-Array importiert. Dieses brauchte ich, da ich ja per Mausklick aus dem Array die Sätze holen möchte und in das <p>-Tag schreiben möchte. Ich habe den Inhalt dieser Funktionalität in eine Klasse geschrieben um die komplette Klasse auf einmal exportieren zu können. Als erstes erstelle ich hier eine private Variable, mit der ich auf das <p>-Tag zugreife, in welches ich dann schlussendlich meinen Satz anzeigen möchte. Ich habe dieses Element ebenso als „HTMLElement“ typisiert. Ebenso habe ich eine Variable für den Index des Arrays initialisiert mit der Zahl null, da das Array bei Stelle null beginnen soll. Ich habe diese Variable als „number“ typisiert. Ich habe als nächstes einen Konstruktor erstellt, welcher zu Beginn leer war. Als nächstes habe ich mit der Funktion initButton() begonnen. Diese Funktion hat den Rückgabewert „void“. Ich habe hier zunächst eine Variable erstellt, mit welcher ich per querySelector auf den Button zugegriffen habe. Das Ganze habe ich als HTMLElement typisiert. Als nächstes habe ich auf die Variable (also auf den Button) ein Klick-Event hinzugefügt. Ich habe per Arrow-Funktion zu Beginn einen Console.log eingebaut, welcher mir den Index anzeigen sollte, da beim Klick ja immer hochgezählt werden sollte. Das funktionierte allerdings noch nicht, da die Funktionalität des Hochzählens des Index noch nicht vorhanden war. Also schrieb ich eine Funktion namens nextIndex(), welche den Rückgabewert „void“ besitzt. In dieser Funktion ist eine if-Bedingung. Die prüft ab, ob der momentane Index kleiner oder größer gleich die Indexgröße des Arrays ist. Wenn also der momentane Index kleiner der Größe des Arrays ist, soll der Index um eins erhöht werden. Wenn der Index die größte Zahl des Arrays erreicht hat (Durch -1 in der Bedingung) soll der Index auf 0 gesetzt werden. Diese Funktion funktioniert allerdings so noch nicht, da sie auch ausgeführt werden muss. Jetzt benötigte ich noch die Funktion, welche den Inhalt des Arrays auch in das <p>-Tag schreibt. Daher habe ich die Funktion onClick() geschrieben, welche ebenfalls den Rückgabewert void besitzt. In dieser Funktion greife ich auf die Variable zu, welche auf das <p>-Tag zugreift. Ich benutze hier nun die Methode .innerHTML und schreibe den Inhalt aus dem Array an genanntem Index in das <p>-Tag. In dieser Methode rufe ich auch die Methode nextIndex() auf um dafür zu sorgen das der Index immer weiter hochzählt pro Klick und dann eben ggf. wieder von vorne beginnt. Zurück zur oben bereits erwähnten Arrow Funktion. Die Arrow Funktion war bis jetzt leer und damit hatte der Klick auf den Button noch keine Aktion. Also rief ich in dieser Funktion die Funktion onClick() auf. Die komplette Funktion initButton() rufe ich im letzten Schritt im Konstruktor auf.

Ich hatte bei der Datei button.ts zahlreiche Probleme. Mein erstes Problem war auf den Button zuzugreifen, da ein einfacher Methodenaufruf nicht funktioniert hat. Dieses Problem konnte ich durch die Verwendung einer Arrow Funktion beheben. Ein weiteres Problem hatte ich, da ich nach dem letzten Index mit einem Aussagesatz immer ein „undefined“ erhalten habe. Dieses Problem konnte ich durch die -1 in der if-Bedingung lösen, sodass das Feld dieser nicht definierte Index nicht mehr in dem <p>-Tag angezeigt wird.

## 4. index.ts

Zu guter Letzt importierte ich die Klasse „buttonClass“ von dem Button Modul in die Datei index.ts und erzeugte einen neue buttonClass. Diesen Schritt muss ich machen, da von hier aus der Compiler bundled. Ohne diesen Schritt hätte die Web-App keine Funktion. Ich hätte hier auch in der Konfigurationsdatei den button.ts als input wählen können, dann hätte ich auf das importieren in die index.ts verzichten können.

# 2. Reflexion

Da ich schon Erfahrungen mit JavaScript vor der Vorlesung hatte und ebenfalls mit Java gearbeitet habe, ist mir das Arbeiten mit TypeScript nicht sonderlich schwergefallen. Allerdings war für mich die größte Hürde an diesem Assignment das Erstellen der Basisdatei. Ich habe das Erstellen per Video nochmal Stück für Stück versucht, jedoch scheiterte ich. Es gab dauerhaft verschiedene Arten von Problemen. Ein Problem war z.B., sich keine bundle.js erstellt hat und dadurch die Funktionen des verwendeten Skripts nicht anschaubar waren auf der Seite. Nach einiger Zeit habe ich aufgegeben und die Datei des Dozenten Josef verwendet.

Ich habe in diesem Projekt viel gelernt. Beispielsweise hatte ich bis jetzt noch nie mit Arrow Funktionen gearbeitet. Das Thema habe ich schonmal gelesen gehabt allerdings habe ich durch die Vorlesung und das weitere Recherchieren die Nützlichkeit verstanden. Ich bin sehr froh, das entdeckt zu haben, da ich ansonsten meinen Code anders aufbauen hätte müssen um das Problem zu fixen.

Ich möchte auch noch kurz festhalten, wieso ich auf die Idee mit dem „Ich hab noch nie“ Spiel gekommen bin. Jedes Mal, wenn man mit Freunden an einem geselligen Abend dieses Spiel spielt hat man ein Problem. Die Benutzungsoberfläche ist entweder sehr schlecht umgesetzt im Punkt Design oder man hat all diese Fragen schonmal gemacht. Und was ich noch viel schlechter finde ist, dass bei 99% (Meine Einschätzung ohne jegliche Wissenschaftliche Stützung) der verfügbaren Apps andauernd Werbung geschalten wird, wenn man kein Geld zahlen möchte. Daher fand ich es super eine einfach modifizierbare „Ich hab noch nie“ Web-App zu erstellen, um all diesen Problemen aus dem Weg gehen zu können.

Für die Zukunft würde ich die Web-App erweitern, wenn ich weiterhin daran arbeiten würde. Ich würde ein Feature einbauen, welches in einem erneuten Array alle Favoritenaussagen abspeichert um diese extern in einem anderen „Favoriten-Ich hab noch nie“ auszuführen. Ebenso würde ich zusätzlich ein Zurück-Button einbauen, welcher es ermöglicht zur vorherigen Fragen zurückzuspringen, falls sich der Nutzer einmal verdrückt.