

Plaudio

1. Einleitung

Plaudio ist eine Progressive-Web-App (PWA), die Sprachaufnahmen und GPS-Location aufzeichnet und nach Nextcloud hochladet. Die Webseite kann selbst Offline Sprachaufnahmen aufnehmen und durch ein IndexedDB auf der Webseite bzw. auf dem Gerät zwischenspeichern. Sobald eine Verbindung besteht, können die Sprachaufnahmen nach Nextcloud upgeloadet werden. Nextcloud ist eine self-hosted Plattform, die es erlaubt, Daten auf einem eigenen Server abzulegen und zu verwalten. Plaudio ist mit Angular erstellt und mit Angular-Material designet. Das Backend ist Dockerized, sodass es mit Docker auf einem Server ausgeführt werden kann.

Übersicht der Funktionen von Plaudio:

- Die Webseite kann nur von Benutzern besucht werden, die einen Account besitzen
- Benutzer müssen, im Backend angelegt und verwaltet werden
- Benutzer können Unternehmen zugeordnet werden, sodass sie die upgeloadeten Sprachaufnahmen der anderen Mitarbeiter sehen und abspielen können
- Plaudio kann über PC oder Mobile Geräte genutzt werden
- Durch ein Mikrofon am Gerät werden die Sprachaufnahmen der Benutzer aufgezeichnet
- Durch ein IndexedDB werden die Sprachaufnahmen auf dem Gerät zwischengespeichert
- Die Sprachaufnahme Funktion kann Offline genutzt werden
- Die Sprachaufnahme kann nach Nextcloud upgeloadet werden. Wenn es Erfolgreich hochgeladen ist, wird die zwischengespeicherte Aufnahme auf dem Gerät gelöscht
- Sprachaufnahmen, die nach Nextcloud upgeloadet wurden, können auf Nextcloud wieder gefunden werden
- Plaudio unterstützt die Sprachen Deutsch und Niederländisch

In diesem Dokument wird die Umsetzung der oberen Funktionen erklärt.

2. Material

Damit die Funktionen umsetzbar sind, wurden verschiedene Pakete für Angular installiert und genutzt. Diese werden in diesem Kapitel aufgelistet.

2.1 Frontend

Das Frontend wurde mit dem Framework Angular entwickelt. Das Design wurde mit Angular-Material umgesetzt.

2.1.1 recordrtc

Damit die Webseite die Sprache aufzeichnet, wird "recordrtc" verwendet. Es bietet Funktionen an, um mit dem Mikrofone das Audio aufzuzeichnen, die Aufnahme zu stoppen, fortzufahren und abzubrechen. Diese Aufgaben können auch mit der nativen Javascript Sprache umgesetzt werden, jedoch konnte die Aufgabe mit "recordrtc" schneller gelöst werden. Zukünftig könnte "recordrtc" mit der nativen Javascript Sprache ersetzt werden.

2.1.2 service-worker

Die Webseite soll auch Offline funktionieren können, weshalb ein Service Worker implementiert wurde. Angular bietet einen Service Worker bereits an, der nur mit NPM installiert werden muss. Wenn ein Nutzer diese Webseite besucht, wird der Service-Worker dafür sorgen, dass der Browser die Webseite ins Gerät zwischenspeichert. Wenn der Benutzer die Website erneut besucht, muss der Browser die Webseite nicht erneut laden und wenn das Gerät offline ist, kann die Webseite weiterhin aufgerufen und genutzt werden, weil die Webseite im Browser zwischengespeichert ist.

2.1.3 dexie

Damit Plaudio nicht abhängig vom Backend ist, um Sprachaufnahmen zu speichern, wird ein IndexedDB verwendet. IndexedDB erlaubt im Webbrowser strukturierte Daten zu speichern. Solange der Cache nicht geleert wird, bleiben die Sprachaufnahmen auf dem Gerät erhalten. Wenn das Gerät offline ist, können weiterhin Sprachaufnahmen im Webbrowser gespeichert werden. Wenn der Benutzer die Sprachaufnahmen nach Nextcloud hochladen möchte, müssen die Sprachaufnahmen manuell ausgewählt und per Knopf hochgeladen werden.

2.1.4 file-saver

Die Sprachaufnahmen können auch auf einem Gerät heruntergeladen werden. Die native Javascript Sprache bietet Funktionen an, die Aufgabe zu lösen, jedoch, damit die Aufgabe schnell gelöst ist, wurde das Paket "file-saver" mit NPM installiert. Zukünftig kann "file-saver" mit der nativen Javascript Sprache ersetzt werden.

2.1.5 jwt-decode

Wenn sich ein Benutzer mit seinem Account anmeldet, dann wird ein JSON Web Token (JWT) an den Benutzer vergeben. Der Token wird mit dem Paket "jwt-decode" decoded.

2.1.6 ngx-translate

Damit Plaudio in verschiedenen Sprachen verwendet werden kann, wird das Paket "ngx-translate" verwendet. In "src/assets/i18n" befinden sich alle Sprachen im JSON Format. In den HTML Dateien wird der Text nicht ausgeschrieben, sondern durch eine Funktion aus den JSON Dateien aus "src/assets/i18n" entnommen. Wenn eine andere Sprache angezeigt werden soll, wird der Text aus einer anderen JSON Datei entnommen und in HTML ausgegeben.

2.2 Backend

Das Backend wurde mit Node.js entwickelt und kann mit Docker aufgesetzt werden. Das Backend verwaltet die Benutzer-Authentifizierung und leitet die Sprachaufnahmen nach Nextcloud weiter.

2.2.1 jwt-decode

Es wird genutzt, um einen Token zu decoden und die Informationen aus dem Token entnehmen zu können.

2.2.2 jsonwebtoken

Das Paket kann JSON Web Token (JWT) erstellen, verifizieren und decoden. Wenn sich ein Benutzer erfolgreich anmelden kann, erhält der Benutzer einen Token, mit dem er/sie sich wieder anmelden kann.

2.2.3 nextcloud-node-client

Sprachaufnahmen werden durch das Paket "nextcloud-node-client" nach Nextcloud hochgeladen.

2.2.4 multer

Damit Sprachaufnahmen vom Frontend zum Backend kommen können, muss die API Dateien im Request erlauben. Es kann mit Express eingestellt werden, jedoch, damit es schneller gelöst ist, wird das Paket "multer" installiert und verwendet. Multer bietet Funktionen an, die die Einstellung übernehmen.

3. Funktionen

In diesem Kapitel wird die Umsetzung der Software erläutert.

3.1 Anmeldung

Beim Besuch der Webseite muss sich der Benutzer zuerst anmelden. Es wird ein Account benötigt. Der Benutzer kann einen Account nur durch den Admin bekommen. Einen eigenen Account können Benutzer nicht erstellen. Im Backend gibt es ein Node.js Programm mit den Namen "registerUser.js". Dieses Programm erlaubt es, neue Benutzer anzulegen. Zusätzlich kann dem Benutzer ein Unternehmen zugeordnet werden. Nach dem Anmelden wird der Benutzer umgeleitet und erhält einen Token, mit dem man sich wieder anmelden kann. Der Token wird mit dem Paket "jsonwebtoken" im Backend erstellt.

Siehe in Frontend:

- src/app/components/login
- src/app/services/authservice

Siehe in Backend:

- app.js

- controllers/auth.controller.js
- registerUser.js

3.2 Sprachaufnahme und GPS Location

Nach der Anmeldung können Sprachaufnahmen direkt aufgenommen und abgespeichert werden. Das Paket "recordrtc" übernimmt die Aufnahme. Durch ein Knopf mit einem Mikrofon Symbol kann die Aufnahme beginnen, ein Knopf mit einem Stopp Symbol kann die Aufnahme stoppen, ein Knopf mit einem Abbrechen Symbol kann die Aufnahme abbrechen und ein Knopf mit einem Senden Symbol kann die Sprachaufnahme abspeichern.

Optional können zusätzlich noch die GPS Koordinaten des Gerätes mitgegeben werden. Dazu muss nur eine Schaltfläche betätigt werden. Die Webseite erhält die GPS-Koordinaten durch natives Javascript.

Der Name der Sprachaufnahme besteht aus dem Datum, in dem die Sprachaufnahme gespeichert wurde. Das Datum ist im ISO 8601 Format. Zusätzlich steht im Dateinamen noch der Name des Benutzers und, wenn die GPS Location aktiv ist, die Latitude und longitude. Die Informationen sind mit "--" getrennt.

Beispiele eines Dateinamen:

- 2022-07-11T152333UTC+2--test.wav
- 2022-07-12T121428UTC+2--test--51.486591-6.526544.wav

Siehe Frontend:

- src/app/components/audio-upload
- src/app/services/audio-upload

Siehe Backend:

- app.js
- controllers/upload.controller.js

3.3 Speichern der Sprachaufnahme mit IndexedDB

Mit dem Paket "dexie" werden die Sprachaufnahmen in einem IndexedDB gespeichert und aufbewahrt, bis die Sprachaufnahmen nach Nextcloud hochgeladen, manuell gelöscht oder der Cache geleert wird.

In "src/app/services/indexedDB" sind alle Funktionen, um "dexie" zu steuern.

Siehe Frontend:

- src/app/components/audio-upload
- src/app/services/indexedDB

3.4 Übersicht der Sprachaufnahmen

Alle erstellten und hochgeladenen Sprachaufnahmen werden in zwei Tabellen dargestellt. Die Sprachaufnahmen auf dem Gerät können abgespielt und gedownloadet werden. Die Sprachaufnahmen in Nextcloud können über einen Button mit dem Namen "Zum Nextcloud

Ordner umleiten" erreicht werden. Dort können die Sprachaufnahmen abgespielt und gedownloadet werden. Die Sprachaufnahmen können die Benutzer in Nextcloud nicht löschen.

Die Tabelle wird durch ein Angular Directive erzeugt. Es kann im Ordner "src/app/directives" gefunden werden.

Mit dem Paket "file-saver" werden die Sprachaufnahmen gedownloadet. Die Aufgabe kann mit nativem Javascript gelöst werden. Zukünftig könnte es mit nativem Javascript ersetzt werden.

Siehe Frontend:

- src/app/components/audio-upload
- src/app/services/audio-upload
- src/app/services/indexedDB

Siehe Backend:

- routes.js
- controllers/upload.controller.js

3.5 Hochladen von Sprachaufnahmen nach Nextcloud

Damit die Sprachaufnahmen nach Nextcloud hochgeladen werden kann, müssen Sprachaufnahmen in der Übersicht ausgewählt werden. Durch einen Knopf mit dem Namen "Hochladen zu NextCloud" werden die ausgewählten Sprachaufnahmen zum Backend gesendet. Dort werden sie zum Nextcloud Server weitergeleitet. Für den Nextcloud Server sind Credentials erforderlich, damit Daten hochgeladen werden können. Die Credentials müssen im Docker Ordner in der Datei "docker-compose.yml" verändert werden.

Siehe Frontend:

- src/app/components/audio-upload
- src/app/services/audio-upload

Siehe Backend:

- routes.js
- controllers/upload.controller.js

Siehe Docker:

- docker-compose.yml

3.6 Offline Funktion

Damit Plaudio Offline Sprachaufnahmen aufnehmen und abspeichern kann, muss die Webseite ohne Internetzugriff erreichbar sein. Dafür wurde ein Service Worker eingerichtet. Mit NPM kann das Paket "service-worker" installiert werden. Weitere Schritte waren nach der Installation nicht erforderlich. Nachdem ein Benutzer Plaudio auf seinem Browser besucht hat, wird die Webseite auf dem Browser zwischengespeichert. Sobald der Benutzer die Website erneut besucht, benötigt der Browser die Webseite nicht erneut zu downloaden. Dies erlaubt es, Datenverkehr zu sparen und die Website Offline zu verwenden. Damit der Benutzer auch Offline authentifiziert wird, wird Offline überprüft, ob der Benutzer ein

Anmelde-Token besitzt. Falls kein gültiger Token vorhanden ist, erhält der Benutzer keinen Zugriff auf das Plaudio. Wie schon erwähnt, wird ein IndexedDB verwendet, um Sprachaufnahmen offline zu speichern. Siehe in Kapitel 3.3.

Plaudio findet heraus, dass es keinen Internetzugriff besitzt, indem es diesen Codeausschnitt verwendet.

```
// check if Online
this.isOffline = !window.navigator.onLine
window.onoffline = (event) => {
  this.isOffline = true
};
window.ononline = (event) => {
  this.isOffline = false
};
```

Der “Hochladen zu NextCloud” Knopf wird im Offline Status deaktiviert und wieder aktiviert, sobald wieder eine Internetverbindung besteht.

Siehe Frontend:

- src/app/components/audio-upload
- src/manifest.webmanifest
- src/ngsw-config.json
- src/app/services/authservice
- src/app/auth.guard.ts

3.7 Verschiedene Sprachen

In den HTML Dateien wird die Klasse aus “src/app/service/translater” verwendet, die es ermöglicht, aus den JSON Dateien aus “src/app/assets/i18n” ein Text zu entnehmen. Wenn die Sprache geändert wird, wird eine andere JSON Datei aus “src/app/assets/i18n” benutzt.

Siehe Frontend:

- src/app/components/audio-upload
- src/app/components/login
- src/app/service/translater
- src/app/assets/i18n