Mini Custom Text-to-Speech App Documentation

Introduction

This comprehensive documentation provides a detailed explanation of the mini custom text-to-speech app built using Angular. Users can select a language, input text, and convert it into speech using pre-recorded audio files. Each code block and method is described for clarity.

Getting Started

Prerequisites:

- Ensure Angular and its CLI are installed on your system.

Code Installation:

- Obtain the app's source code from your repository.

Running the App:

- Open your terminal in the project directory.
- Launch the app with `ng serve --open`.
- Access the app via your web browser at `http://localhost:4200/`.

App Components

`app-header`

- Displays the app's header.

`app-sidebar`

- Provides a sidebar for app navigation.

`app-audio-player`

- The main component responsible for filtering and playing audio based on user input.

Using the App

Language Selection:

1. At the top of the app's homepage, you'll find a "Select Language" dropdown. Choose from available languages such as Igbo, Yoruba, and Hausa.

Typing Words

2. Beneath the language selection, locate the "Type words to speech" input field. Input the words you wish to convert into speech.

3. As you type, the app will attempt to match your input with predefined phrases in the selected language.

4. You can clear your input by clicking the "x" icon next to the input field.

Playing Audio

5. When a match is found for your input, a "Speak" button will appear. Click it to hear the corresponding audio for the matched phrase.

6. The audio will play, and you will see an animated speaker icon and the matched phrase displayed below it.

No Search Result?

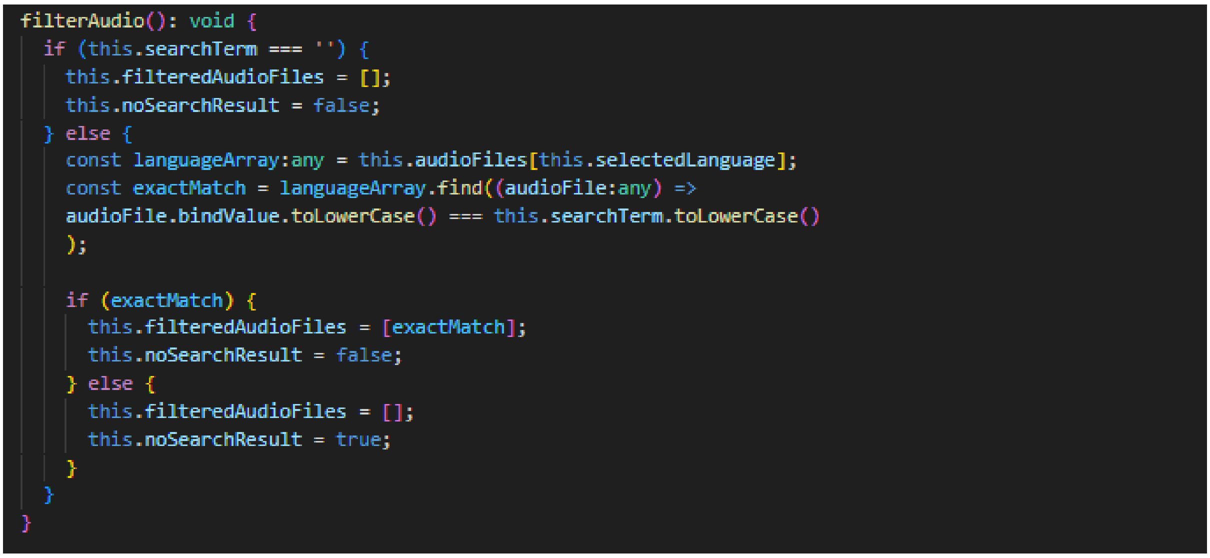
7. If no matching phrase is found for your input, the app will display a message indicating "No Audio found for the searched keyword."

App Logic

`AudioPlayerComponent`

This is the central component responsible for the app's core logic.

`filterAudio()`



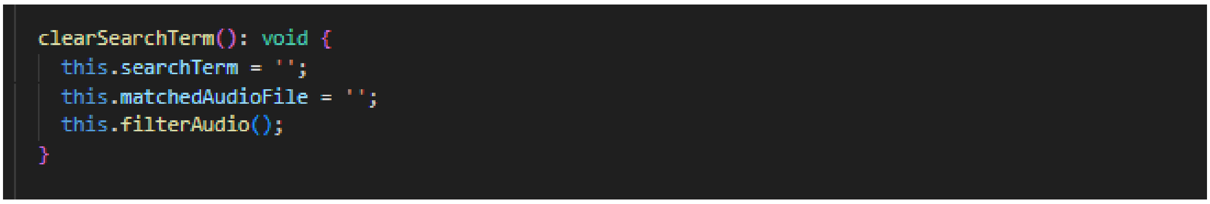
```
filterAudio(): void {
  if (this.searchTerm === '') {
    this.filteredAudioFiles = [];
    this.noSearchResult = false;
  } else {
    const languageArray:any = this.audioFiles[this.selectedLanguage];
    const exactMatch = languageArray.find((audioFile:any) =>
    audioFile.bindValue.toLowerCase() === this.searchTerm.toLowerCase()
    );

    if (exactMatch) {
      this.filteredAudioFiles = [exactMatch];
      this.noSearchResult = false;
    } else {
      this.filteredAudioFiles = [];
      this.noSearchResult = true;
    }
  }
}
```

- This method filters audio based on the selected language and user-input words.

- If the input field is empty, it resets the audio display.

- If a match is found, it displays the matched phrase and allows users to click "Speak" to play the corresponding audio.

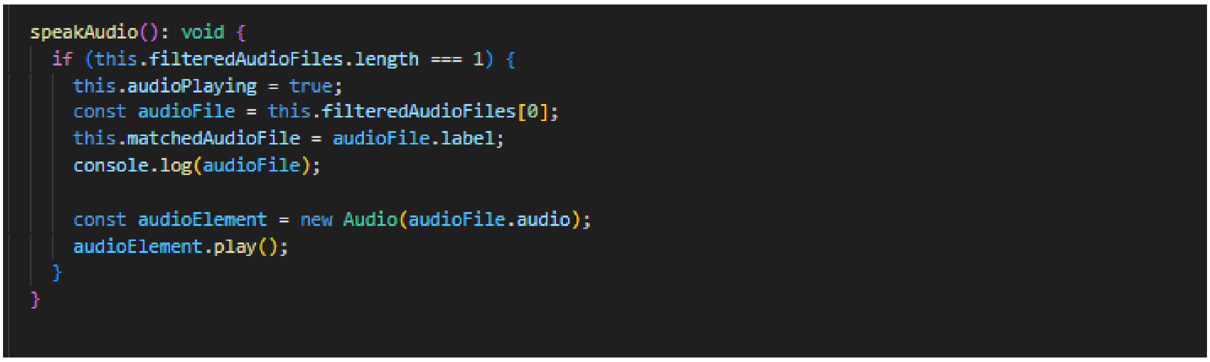
`clearSearchTerm()`



```
clearSearchTerm(): void {
  this.searchTerm = '';
  this.matchedAudioFile = '';
  this.filterAudio();
}
```

- Clears the search input field and resets the audio display when the "x" icon is clicked.

`speakAudio()`



```
speakAudio(): void {
  if (this.filteredAudioFiles.length === 1) {
    this.audioPlaying = true;
    const audioFile = this.filteredAudioFiles[0];
    this.matchedAudioFile = audioFile.label;
    console.log(audioFile);

    const audioElement = new Audio(audioFile.audio);
    audioElement.play();
  }
}
```

- Initiates audio playback when the user clicks "Speak" after a match is found.

- Displays the matched phrase and plays the associated audio.

Audio Data

`AudioService`

This service class provides audio files for different languages. Each audio file is linked to a specific phrase in the selected language.

`getAudioFiles()`

- Retrieves and returns the audio files for the selected language.

Additional Information

- Audio files are stored in the `assets/audiorecords` directory.

- Custom words are stored in the assets/word-document` directory. (As also listed below).

List of Words documented and speech synthesized for each of the three languages:

1. Good morning
2. Good afternoon
3. Good night
4. Welcome
5. Thankyou
6. Eat
7. Water
8. Run
9. Goodbye
10. Go
11. Come

- The app currently supports three languages (Igbo, Yoruba, and Hausa), but you can expand its language support by extending the `audioFiles` object in the `AudioService`.

This documentation offers a comprehensive understanding of the app's components and methods, enabling you to use it effectively for text-to-speech conversion in the supported languages. Enjoy your custom text-to-speech experience!