Voice Recorder API User Guide/ Playbook

This API allows you to integrate speech recognition functionality into your web applications, making it possible for users to input text using their voice.

To get started with this API, follow the steps below:

1. Include the following script in your HTML file:

**<script src="https://cdnjs.cloudflare.com/ajax/libs/webkitSpeechRecognition/0.5.0/webkitSpeechRecognition.js"></script>**

1. Create a new instance of the **webkitSpeechRecognition** object:

**const recognition = new webkitSpeechRecognition();**

1. Set the language of the recognition:

**recognition.lang = 'en-US';**

1. Add an event listener for the **result** event:

**recognition.addEventListener('result', e => { const transcript = e.results[0][0].transcript; console.log(transcript); });**

1. Start the recognition process:

**recognition.start();**

API Reference:

Here are the methods and properties available in the **webkitSpeechRecognition** object:

Methods:

* **start()**: Starts the recognition process.
* **stop()**: Stops the recognition process.
* **abort()**: Aborts the recognition process.

Properties:

* **continuous**: A Boolean indicating whether the recognition should continue after the user stops speaking.
* **interimResults**: A Boolean indicating whether interim results should be returned (i.e., results that are not final).
* **lang**: The language of the recognition (e.g., 'en-US', 'fr-FR', etc.).
* **maxAlternatives**: The maximum number of alternatives to consider for each recognition result.

Examples:

Here are some examples of how you can use this API:

Example 1: Basic Speech Recognition

This example listens for speech and logs the transcript to the console.

**const recognition = new webkitSpeechRecognition(); recognition.lang = 'en-US'; recognition.addEventListener('result', e => { const transcript = e.results[0][0].transcript; console.log(transcript); }); recognition.start();**

Example 2: Continuous Speech Recognition

This example listens for speech continuously and logs the transcript to the console.

**const recognition = new webkitSpeechRecognition();**

**recognition.lang = 'en-US';**

**recognition.continuous = true;**

**recognition.interimResults = true;**

**recognition.addEventListener('result', e => {**

**const transcript = e.results[0][0].transcript;**

**console.log(transcript);**

**});**

**recognition.start();**

Example 3: Speech Recognition with Multiple Alternatives

This example listens for speech and returns up to 5 alternative transcriptions for each result.

**// Create a new instance of webkitSpeechRecognition**

**const recognition = new webkitSpeechRecognition();**

**// Set the language of the recognition to English (United States)**

**recognition.lang = 'en-US';**

**// Add an event listener for the "result" event**

**recognition.addEventListener('result', e => {**

**// Get the first result and its transcript**

**const transcript = e.results[0][0].transcript;**

**// Output the transcript to the console**

**console.log(transcript);**

**});**

**// Start the recognition process**

**recognition.start();**

Conclusion:

With this API, you can easily add voice recognition capabilities to your web applications.