Lingua Workbench
Personal English Listening Study Platform

Contents

Preface

About This Guide

This documentation provides comprehensive guidance for using Lingua Workbench, a personal language learning platform for studying spoken English from authentic audio sources.

Who Should Use This Guide

This guide is intended for:

- **Language learners** who want to improve their English listening skills through TV shows and movies
- **Developers** who want to set up and customize the platform
- **Contributors** who want to understand the system architecture

How This Guide is Organized

This guide is divided into the following parts:

- **Getting Started** – Installation and first-run instructions
- **User Guide** – Step-by-step procedures for using all features
- **Reference** – Lookup information for controls, tags, and technical details
- **Troubleshooting** – Solutions to common problems
- **Appendix** – Additional reference materials

Conventions Used

This documentation uses the following conventions:

- **Bold text in boxes** indicates UI buttons or controls
- `Monospace text` indicates code, commands, or file paths
- **Bold text** indicates important terms or emphasis

Notices

Copyright

Trademarks

- Vue.js is a trademark of Evan You.
- Django is a registered trademark of the Django Software Foundation.
- OpenAI and Whisper are trademarks of OpenAI, Inc.
- All other trademarks are the property of their respective owners.

Third-Party Services

This application uses the following third-party services:

- **OpenAI API** – For phonetic analysis and dictionary features
- **Whisper API** – For speech-to-text transcription

Usage of these services is subject to their respective terms of service and pricing.

Disclaimer

Revision History

This section documents the revision history of Lingua Workbench.

Version	Date	Author	Changes
1.0	2026-01-08	Edith Tang	• Initial release • Audio management (upload, chunk, slice) • Whisper transcription integration • AI phonetic analysis (Sound Script) • Dictionary with bilingual examples • Playback speed control • Favorite marking • Time adjustment arrows • Independent slice deletion

Part I. Getting Started

Chapter 1. Product Overview

Lingua Workbench is a personal language learning platform for studying spoken English.

What is Lingua Workbench?

Lingua Workbench helps you analyze and study spoken English from audio sources like TV shows and movies. It provides tools for:

- Uploading and managing audio files organized by drama/season/episode
- Creating precise audio slices from longer recordings
- AI-powered transcription and phonetic analysis
- Context-aware dictionary lookups

Key Features

The platform includes:

- **Audio Slicer** - Interactive waveform selection and region management
- **Sound Script Analysis** - AI-generated phonetic breakdown of phrases
- **Dictionary Integration** - Bilingual definitions with contextual examples
- **Playback Controls** - Variable speed, loop, and region playback

Target Audience

This tool is designed for English learners who want to master natural spoken English through authentic audio sources.

Chapter 2. Installing Lingua Workbench

Set up the development environment for Lingua Workbench.

Before you begin, ensure you have:

- Node.js 20+ or 22+
- Python 3.11+
- ffmpeg (for audio processing)
- Git

Lingua Workbench consists of a Vue 3 frontend and Django backend. Both need to be set up separately.

Create environment files with your API keys.

Both frontend and backend are now ready to run.

Chapter 3. Running for the First Time

Start the development servers and access the application.

Complete the installation *(on page 9)* first.

1. Start the backend server:

```
cd backend

venv\Scripts\activate

python manage.py runserver
```

The Django server starts on `http://localhost:8000`

2. In a new terminal, start the frontend:

```
cd frontend

npm run dev
```

The Vite dev server starts on `http://localhost:5173`

3. Open your browser and navigate to `http://localhost:5173`

You should see the Lingua Workbench home page. You can now start uploading audio and creating slices.

Part II. User Guide

Chapter 1. Audio Management

Audio Workflow Overview

Understanding how audio is organized and processed in Lingua Workbench.

Audio Hierarchy

Audio in Lingua Workbench is organized in three levels:

1. **Source Audio** - The original uploaded file, organized by Drama → Season → Episode
2. **Audio Chunks** - Automatically created 5-minute segments of the source audio
3. **Audio Slices** - User-created regions within chunks for focused study

Processing Flow

When you upload an audio file:

1. The file is stored and associated with a drama/episode
2. It is automatically split into 5-minute chunks
3. You can then browse chunks and create slices for study

Uploading Source Audio

Upload an audio file from a TV show or movie.

You need an audio file in MP3 or WAV format.

1. Navigate to the Audio Upload page.
2. Select or create a Drama (e.g., "Friends").
3. Enter the Season and Episode numbers.
4. Optionally add an episode title.
5. Click **Choose File** and select your audio file.
6. Click **Upload**.

The audio is uploaded and automatically chunked. You can now browse the chunks.

Browsing Audio Chunks

View and select audio chunks for slice creation.

1. Navigate to the Audio Workbench page.
2. Use the filter to select a drama or episode.
3. Click on a chunk card to open it in the Audio Slicer.

The waveform is displayed and you can begin creating slices.

Chapter 2. Creating Slices

What is an Audio Slice?

Understanding audio slices and their role in language learning.

Definition

An *audio slice* is a user-defined region within an audio chunk. Each slice represents a meaningful utterance, sentence, or phrase that you want to study.

Slice Components

Each slice contains:

- **Time boundaries** - Start and end times within the chunk
- **Transcription** - The text content (via Whisper or manual input)
- **Highlights** - Selected text portions for phonetic analysis
- **Favorite flag** - Mark important sentences for review

Best Practices

- Keep slices focused on single sentences or short exchanges
- Use the time adjustment arrows to fine-tune boundaries
- Star slices that contain challenging pronunciations

Creating a Slice

Select a region on the waveform to create an audio slice.

You must have an audio chunk open in the Audio Slicer.

1. Click and drag on the waveform to select a region.
   The selected region is highlighted in blue.

2. Release the mouse button to create the slice.
   A new SliceCard appears in the panel below.

3. **Optional:** Fine-tune the boundaries using the ◀ and ▶ arrows on the timestamp.
   Each click adjusts the time by 0.5 seconds.

The slice is created locally. Use **Save All Changes** to persist to the server.

Transcribing a Slice

Use AI to automatically transcribe the audio in a slice.

1. Locate the slice you want to transcribe.

2. Click the **Transcribe** button (microphone icon).
   The button shows a loading indicator while processing.

3. Wait for the transcription to complete.
   The transcribed text appears in the slice card.

4. **Optional:** Click the edit button to correct any transcription errors.

The slice now contains the transcribed text, ready for highlighting and analysis.

Editing Slice Text

Manually edit the transcription text of a slice.

1. Click the edit button (pencil icon) at the bottom-right of the text area.
   The text becomes editable in a textarea.

2. Make your changes to the text.

3. Click the save button (floppy disk icon) to save changes.

4. **Optional:** Click the cancel button (X icon) to discard changes.

Deleting a Slice

Remove a slice from your collection.

Deleting a saved slice immediately removes it from the server. This action cannot be undone.

Click the delete button (trash icon) on the slice card.

For saved slices, the deletion is immediate. For unsaved slices, only the local region is removed.

Chapter 3. Phonetic Analysis

Sound Script Analysis

Understanding AI-generated phonetic analysis.

What is Sound Script?

Sound Script is an AI-generated phonetic breakdown of spoken English phrases. It analyzes how words are actually pronounced in connected speech, identifying:

- **Reductions** - Vowels weakening to schwa sounds
- **Linking** - Words connecting across boundaries
- **Assimilations** - Sounds changing due to neighboring sounds
- **Deletions** - Sounds that are dropped (h-deletion, t-deletion)

Sound Display Notation

The `sound_display` shows pronunciation using:

- Phonetic spelling (e.g., "wuh-duh-we" for "what do we")
- Square brackets for ghost/silent sounds: `[t]uh`
- Hyphens to show syllable/word boundaries

Ruby Text Display

The UI displays pronunciation above the original text using Ruby annotation, making it easy to compare written and spoken forms.

Highlighting Text for Analysis

Select a phrase to analyze its pronunciation.

The slice must have transcribed text.

1. Select text in the slice by clicking and dragging.
   A highlighter icon appears near your selection.

2. Click the highlighter icon to create a highlight.
   The text is highlighted and the Highlight Editor opens below.

The highlight is created and you can now request AI analysis.

Getting AI Analysis

Request phonetic analysis for a highlighted phrase.

You must have a highlight selected.

1. With a highlight active, click the **Analyze** button in the Highlight Editor.
   The AI processes the phrase and returns analysis.

2. Review the phonetic tags and sound display.

The analysis shows phonetic breakdown with Ruby text annotations above the original text.

Editing Sound Display

Correct or refine the AI's pronunciation analysis.

Sometimes the AI makes mistakes in phonetic analysis. You can manually correct the sound_display values.

1. Switch to Sound Display mode by clicking the mode toggle button.

2. Click on a segment to select it for editing.
   The segment's sound_display appears in the input field.

3. Edit the pronunciation text.

4. Click **Save** to apply changes.

Chapter 4. Dictionary

Looking Up a Phrase

Get dictionary definitions for a highlighted phrase.

1. With a highlight active, click the **Dictionary** button.
2. Review the definition and example sentences.

The dictionary panel shows bilingual definitions with contextual examples.

Refreshing Example Sentences

Generate new example sentences for a dictionary entry.

In the Dictionary panel, click the **Refresh** button.

The AI generates new contextual examples in both English and Chinese.

Part III. Reference

Chapter 1. UI Reference

SliceCard Controls Reference

Reference for all controls available on a SliceCard.

Timestamp Bar

Icon	Action	Description
◀	Adjust Start	Move start time 0.5s earlier
▶	Adjust End	Move end time 0.5s later
★	Toggle Favorite	Mark/unmark as favorite

Action Buttons

Button	Function
Transcribe	Send audio to Whisper for transcription
Delete	Remove the slice (immediate for saved slices)
Edit	Enter text editing mode

Playback Controls Reference

Reference for audio playback controls.

Speed Control

Available playback speeds: 0.5x, 0.75x, 1.0x (normal), 1.25x, 1.5x

Loop Mode

When enabled, the active region plays repeatedly until stopped.

Region Playback

Click on a SliceCard to play that region only.

Editor Modes Reference

Reference for Highlight Editor modes.

AI Note Mode

Displays phonetic tags with their notes. Click a tag to edit its annotation.

Sound Display Mode

Displays script segments in format: [original] sound_display. Click a segment to edit its pronunciation.

Mode Toggle

Click the mode toggle button to switch between the two modes.

Chapter 2. Phonetic Notation

Phonetic Tags Reference

Reference for phonetic tag types used in analysis.

Tag	中文	Description
reduction	弱化	Vowel reduced to schwa
linking	连读	Consonant-vowel linking across words
assimilation	同化	Sound changes due to adjacent sounds
h_deletion	H删除	Initial /h/ dropped in function words
flap_t	闪音T	T/D flapped between vowels
glottal_stop	喉塞音	T replaced with glottal stop
elision	省略	Sounds deleted entirely

Ghost Sounds Notation

Understanding the notation for silent or omitted sounds.

Bracket Notation

Sounds enclosed in square brackets `[x]` represent "ghost sounds" - phonemes that are weakened or omitted in natural speech.

Display

Ghost sounds are displayed with strikethrough styling and reduced opacity in the UI.

Examples

Notation	Meaning
`[h]e`	The /h/ in "he" is silent
`wha[t]`	The /t/ in "what" is dropped
`uh[p]`	The /p/ is barely pronounced

Part IV. Troubleshooting

Chapter 1. Transcription Fails or Returns Empty

Resolve issues with Whisper transcription.

Symptom: Clicking the Transcribe button results in empty text or an error.

Possible Causes and Solutions:

1. Check that the Whisper service is running.
   Ensure the Whisper API container is started.

2. Verify the audio region is not empty.
   The selected region must contain audible speech.

3. Check browser console for errors.
   Network or API errors will appear in the console.

4. Ensure the audio format is supported (MP3, WAV).

Chapter 2. AI Analysis Errors

Resolve issues with phonetic analysis.

Symptom: Analysis fails or returns unexpected results.

Troubleshooting Steps:

1. Check your OpenAI API key is valid and has credits.
2. Ensure the highlighted text is meaningful (not just punctuation).
3. Try a shorter phrase if the text is very long.
4. Check the Django server logs for API errors.

Keyboard Shortcuts

Quick reference for keyboard shortcuts.

Available Shortcuts

Shortcut	Action
Space	Play/Pause audio
Esc	Close active highlight editor

> ✎ **Note:**
> More shortcuts may be added in future versions.

Technology Stack

Technical details about the technologies used.

Frontend

Technology	Purpose
Vue 3	UI framework with Composition API
TypeScript	Type-safe JavaScript
Vite	Build tool and dev server
Element Plus	UI component library
Tailwind CSS	Utility-first CSS
WaveSurfer.js	Audio waveform visualization
Pinia	State management
Axios	HTTP client

Backend

Technology	Purpose
Django	Web framework
Django REST Framework	API development
SimpleJWT	JWT authentication
LangChain	LLM orchestration
OpenAI API	AI analysis and dictionary

Services

Service	Purpose
Whisper API	Speech-to-text transcription
OpenAI GPT-4	Phonetic analysis and examples