

BÁO CÁO TIẾN ĐỘ DỰ ÁN: GOOGLE TEXT TO SPEECH API

Người thực hiện: Nguyễn Đỗ Anh Tài

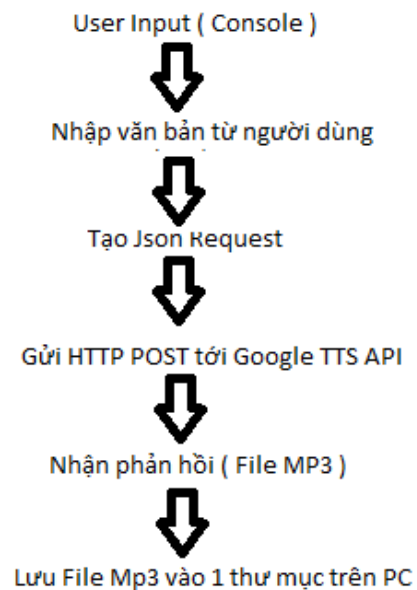
Ngày bắt đầu: 3/12/2024

Ngày gửi báo cáo lần 1: 8/12/2024

Giới thiệu về dự án Google Text-to-Speech API:

Dự án là một công cụ chuyển văn bản thành giọng nói (Text-to-Speech) sử dụng Google Text-to-Speech API và được phát triển bằng ngôn ngữ C. Ứng dụng cho phép người dùng nhập văn bản, chọn giọng nói, ngôn ngữ, tốc độ đọc sau đó chương trình sẽ xử lý và đầu ra là dạng âm thanh với định dạng là MP3. Người dùng có thể lưu hoặc phát lại âm thanh tùy mục đích sử dụng, Dự án tích hợp với Google API để chuyển đổi văn bản thành giọng nói tự nhiên và dễ nghe.

I. Sơ đồ khái quát dự án:



Mô tả sơ đồ:

1. Người dùng nhập văn bản từ **console**.
2. Ứng dụng xử lý văn bản, tạo một yêu cầu JSON chứa văn bản và thông số (ngôn ngữ, giọng đọc, tốc độ đọc).
3. Yêu cầu JSON được gửi qua HTTP POST tới Google Text-to-Speech API bằng **libcurl**.
4. API trả về phản hồi dưới dạng dữ liệu nhị phân (MP3).
5. Dữ liệu được lưu thành file MP3 trên thư mục được cấu hình sẵn.

II. Công nghệ cần thiết:

1. Ngôn ngữ lập trình

- C :
 - + Mã nguồn chính của ứng dụng.
 - + Dùng để xây dựng logic xử lý (nhận input, gọi API, xử lý output).

2. Trình biên dịch (Compiler)

- GCC (GNU Compiler Collection):
 - + Biên dịch mã C thành tệp thực thi.
 - + Hỗ trợ các tùy chọn liên kết với thư viện ngoài (libcurl, json-c).
- Các tùy chọn khác:

3. Thư viện hỗ trợ

- Libcurl:
 - + Hỗ trợ gửi yêu cầu HTTP(S) đến API.
 - + Giao tiếp với Google Text-to-Speech API.
- Json-c:
 - + Hỗ trợ xử lý dữ liệu JSON.
 - + Tạo yêu cầu JSON gửi API và phân tích phản hồi.
- Thư viện chuẩn C:
 - + <stdio.h>: Nhận input từ console, ghi output vào file.
 - + <stdlib.h>: Xử lý lỗi, cấp phát bộ nhớ động.
 - + <string.h>: Xử lý chuỗi.
 - + <time.h>: (tùy chọn) Thêm timestamp cho file đầu ra.

4. Công cụ phát triển

- Visual Studio Code:
 - + IDE hỗ trợ lập trình C với các tiện ích như C/C++ Extension và Code Runner.
 - + Nơi chạy và kiểm tra chương trình (Powershell, CMD, terminal).

5. Google Text-to-Speech API

- API Key:
 - + Cần một API Key hợp lệ từ Google Cloud.
- Sử dụng:
 - + Gửi yêu cầu HTTP POST với JSON payload.
 - + Nhận phản hồi là tệp âm thanh (MP3).

6. Hệ điều hành

- Windows:
 - + Cài GCC, libcurl, json-c đầy đủ.
 - + Thư mục lưu trữ file đầu ra cần quyền ghi để lưu file âm thanh.

7. Debugging và Testing Tools

- Debugger của Visual Studio Code: Kiểm tra lỗi runtime, logic.
- Curl Command-line (tuỳ chọn): Kiểm tra yêu cầu HTTP trước khi tích hợp.
- JSON Validator: Đảm bảo JSON đúng cú pháp.

Tóm tắt công nghệ cần thiết

Thành phần	Công cụ hoặc Thư viện
Ngôn ngữ lập trình	C
Compiler	GCC, MinGW
Thư viện HTTP	libcurl
Thư viện JSON	json-c
IDE	Visual Studio Code
Terminal	Command Prompt, Powershell
API	Google Text-to-Speech API

III. Các giai đoạn thực hiện:

Giai đoạn	Công việc	Thời gian (dự kiến)
Giai đoạn 1	Chuẩn bị môi trường phát triển: <ul style="list-style-type: none">- Cài đặt GCC, Visual Studio Code, libcurl, json-c.- Kiểm tra môi trường hoạt động.- Cấu hình liên kết thư viện.	3 ngày
Giai đoạn 2	Phân tích và lập kế hoạch: <ul style="list-style-type: none">- Xác định input/output của ứng dụng.- Nghiên cứu tài liệu Google TTS API và yêu cầu JSON.	2 ngày
Giai đoạn 3	Phát triển: <ul style="list-style-type: none">- Tạo chương trình cơ bản để nhận input từ console.- Sử dụng json-c để tạo JSON request.- Gửi HTTP POST với	2-3 ngày

	libcurl. - Nhận phản hồi và lưu dữ liệu vào file MP3.	
Giai đoạn 4	Kiểm thử: - Kiểm tra từng phần (gửi request, nhận phản hồi, lưu file). - Kiểm tra tích hợp toàn bộ quy trình. - Xử lý các lỗi phát sinh (mạng, JSON, ghi file).	1-2 ngày
Giai đoạn 5	Hoàn thiện và triển khai: - Tối ưu mã nguồn và tổ chức thành các hàm rõ ràng. - Cung cấp tệp thực thi và hướng dẫn sử dụng. - Đảm bảo môi trường triển khai đầy đủ.	1 ngày

IV. Tiến độ thực hiện:

Dự án đang đến giai đoạn: Đầu giai đoạn 2

- Các công việc đã hoàn thành:
- Trong giai đoạn 1:
 - + Chuẩn bị môi trường phát triển:
 - + Cài đặt GCC, Visual Studio Code, libcurl
 - + Kiểm tra môi trường hoạt động.
 - + Cấu hình liên kết thư viện.
 - ➔ Còn thiếu: Thư viện Jsion
 - ➔ Khó khăn trong giai đoạn 1:
 - Cài sai MinGW cổ điển, dự án cần cài MinGW-64 (đã khắc phục)
 - Thư viện libcurl không hoạt động: sai lỗi đường dẫn đến thư viện, cài biến môi trường Path bị lỗi, cấu hình liên kết thư viện trong task.json, launch.json sai (đã khắc phục)
- Trong giai đoạn 2:
 - + Phân tích và lập kế hoạch: Xác định input/output của ứng dụng.