**TEXT TO SPEECH WEB APPLICATION**

Group member : Phạm Cao Bằng ( HE130602 )

Trần Mạnh Cường

Trần Phúc Hưng

Nguyễn Đình Thăng

*Abstract*—This paper show my text to speech web application that base on FPTAI [1] text to speech API. This application written in Java Platform, Enterprise Edition (JavaEE) [2] and Python [3]

Keywords—text to speech, FPTAI, JavaEE, Python

# Introduction

Nowadays, computer can do a lot of thinks that we can’t imagine in the past. They can play chess, talk like a human, help our predict something base on our own data, etc. Text to speech is one of the most important field that should be applied. This technic can be use for blindness, etc to help them read without looking in paper. In this papers I will focus on this interesting task. I will build a basic web application which can generate an audio file base on our text input.

To understand this topic I recommend that you should have a basic knowledge about JavaEE, Python and APIs. You need to know what is API and how it work. You also need to know JavaEE and Python because my web application written in Java and use Python backend to connect to FPTAI to get our data.

# Methodology

## Requirement

* JavaEE IDE (Netbeans) [4] and Java Development Kit 1.8 (jdk1.8) [5]
* Python 2 or 3
* requests library ( You can run “***pip install requests”*** on command prompt to install this library )
* Basic knowledge about JavaEE, you should focus on

The Web Tier [6]

* Basic knowledge about Python you an study Python tutorial [7] before begin

## What is Application Program Interface(API)

API [8] is an interface or communication protocol between a client and a server intended to simplify the building of client-side software. It has been described as a “contract” between the client and the server, such that if the client makes a request in a specific format, it will always get a response in a specific format or initiate a defined action.

An API maybe for a web-based system, operating system, database system, computer hardware, or software library

There are many forms of APIs but often include specifications for routines, data structures, object classes, variables, or remote calls. Portable Operating System Interface (POSIX) [9], Windows API [10] and Advanced SCSI Programming Interface (ASPI) [10] are examples of different APIs’s forms

API achitecture :

*Picture 1 : API’s architecture*

## Client-server model

*Picture 2: Client-server model*

Client-Server model [11] is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients. Often clients and servers communicate over a computer network on separate hardware, but both client and server may reside in the same system. A server host runs one or more server programs, which share their resources with clients. A client does not share any of its resources, but it requests content or service from a server. Clients therefore initiate communication sessions with servers, which await incoming requests. Examples of computer applications that use the client–server model are Email, network printing, and the World Wide Web [12].

In general, a service is an abstraction of computer resources and a client does not have to be concerned with how the server performs while fulfilling the request and delivering the response. The client only has to understand the response based on the well-known application protocol, i.e. the content and the formatting of the data for the requested service.

Clients and servers exchange messages in a request–response messaging pattern. The client sends a request, and the server returns a response. This exchange of messages is an example of inter-process communication. To communicate, the computers must have a common language, and they must follow rules so that both the client and the server know what to expect. The language and rules of communication are defined in a communications protocol. All client-server protocols operate in the application layer. The application layer protocol defines the basic patterns of the dialogue. To formalize the data exchange even further, the server may implement an application programming interface (API). The API is an abstraction layer for accessing a service. By restricting communication to a specific content format, it facilitates parsing. By abstracting access, it facilitates cross-platform data exchange.

A server may receive requests from many distinct clients in a short period of time. A computer can only perform a limited number of tasks at any moment, and relies on a scheduling system to prioritize incoming requests from clients to accommodate them. To prevent abuse and maximize availability, server software may limit the availability to clients. Denial of service attacks are designed to exploit a server's obligation to process requests by overloading it with excessive request rates.

## Main Idea

When we work with an API, we had to send a request to API server, our request include all information that need for server to query from database and send response back.

So, first things we need are all data required to send a request to FPT’s API. To do that I build a webpage to collect that data from users. After we have data, we need something to send a request to the API, my python code will do that. It will read data are already collected from the webpage and send request to API. My python code will do the next part, it will read data are collected from webpage and send a request to API server. It also read the response from API server and return the link that contain a url link of my mp3 file. Finally, webpage use this link to display the mp3 file in the webpage.

My idea is pretty sample, there are 2 main parts that we need to build are a webpage using java and a request-response hander using python. We will do that in some upcoming steps.

## File structure in my project

*Picture 2: File structure*

My project contain 2 main part are webpage writed with java and request-response handle writed with python. As we know in the previous, my java code will collect data and my python code is request-response hander.

Other files are text files contains all information to handle request and response in my project and display to webpage.

## Building a webpage (JavaEE)

*Picture 3: Homepage screen*

First, we start with a home screen. This webpage contain a text box, you must enter your name in the text box to get start with my Text to Speech project. When you clicked “Let’s Start”, my homepage will redirect you to the main screen of my project.

My homepage will receive your name and greeting you first time when you come to my website.

Picture 4 show my main page that contain all input field that we need to send a request to FPTAI. This is basic step if you have a little kowledge about JavaEE. Here is my sample webpage written in Java

*Picture 4: Main page*

When you click “Tạo file âm thanh” my application will perform the following action:

* First it take data from my webpage and send to my servlet. Data include:
* Text we need to speech.
* Person voice ( Here I have 6 different voice that we can request ).
* Speed ( Here I have 6 levels of speed ).
* Prosody
* After my serlet receive data from webpage, it will write each data to a text file : text.txt, voice.txt, speed.txt, prosody.txt.

Servlet run my python program using command prompt to send a request to FPTAI and write url link to my mp3 file in **url.txt** file.

* Servlet read the **url.txt** file and embed to my webpage.

This process require internet connection to complete so make sure you have internet connection before you start.

## API request,respone using Python

We already received data from my webpage in the previous part. The next thing that we need to do is connect to API and take data that we needed. In my project I use python with request libraby to send request and handle respone from FPT’s API.

import requests

api\_key = "952acd5968644894b16e8b51e89bc165"

voice = open('voice.txt','r').read()

speed=open('speed.txt','r').read()

prosody = open('prosody.txt','r').read()

text = open('text.txt','r',encoding="utf-8").read()

url="http://api.openfpt.vn/text2speech/v4?api\_key=

{}&voice={}&speed={}&prosody={}"

.format(api\_key, voice, speed, prosody)

response = requests.post(url, data=text.encode('utf-

8'),headers={'voice':voice,'speed':speed,'prosody':prosody})

response = response.json()

link = response['async']

file = open("url.txt", "w")

file.write(link);

file.close()

Note that you should write request and url in one line because of python syntax

This is sample request in Python that we use to send a request to FPTAI’s API. We have 5 parameter that need to be placed in python request are voice, speed, prosody, text, and a very important things that is api key. If you don’t have api key you should register in ***fpt.ai*** as a developer and create your own api key

{"async":"https://s3-ap-southeast-

1.amazonaws.com/text2speech-

v4/male.0.pro.4b5b15285847e83acbb3beb

945434453.mp3",

"error": 0,

"message": "The content will be returned after a

few seconds under the async link.",

"request\_id":"4b5b15285847e83acbb3beb945434453"

}

Above is the sample response success when we send request to the API

Python code will send a request and we will receive a json object that contain link to mp3 file. Python code will read that link and write to file **url.txt** ( We can see in Picture 3 )

## G.Measure

There are one thing that I wonder is whether the response time depends on size of our input sentence. After run my project I measure time needed when server send response each number of words and I have the following result.

*Picture 5:Response time*

As we can see, in the picture above we can see that our response time might not depend on our text input size. It depend on another things such as connection, API server problem, etc…

Another things that I wonder is how many host that my request walk through before it reach FPT’s API. After tracert to FPTAI’s server I receive the following result

We can not tracert to FPTAI’s API server after 30 hops. I think that to connect to this API we need all required field such as api key and all data need to send request

# Conclusion

In this report I already show you what is API and how does this work. API is a powerfull technic that can applied in many field and help human to build a a lot of amazing stuff. In my project I focus on text to speech API and build a web application to speech our text input. As I told in the introduction, this kind of application can help many people such as blindness, etc. This can be applied for teaching student because my project quite simple to understand.

REFERENCES

[1] FptAI webpage

<https://fpt.ai/>

[2] Java Platform, Enterprise Edition (JavaEE)

<https://en.wikipedia.org/wiki/Java_Platform,_Enterprise_Edition>

[3] Python

<https://en.wikipedia.org/wiki/Python_(programming_language)>

[4] Netbean IDE

<https://netbeans.org/>

[5] Java Development Kit 1.8- jdk 1.8

<https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

[6] JavaEE tutorial The Web Tier <https://docs.oracle.com/javaee/7/tutorial/partwebtier.htm#BNADP>

[7] Python tutorial <https://docs.python.org/3/tutorial/index.html>

[8] Application Program Interface (API)

<https://en.wikipedia.org/wiki/Application_programming_interface>

[9] Portable Operating System Interface (POSIX)

<https://en.wikipedia.org/wiki/POSIX>

[10]Windows API

<https://en.wikipedia.org/wiki/Windows_API>

[10] Advanced SCSI Programming Interface (ASPI)

<https://en.wikipedia.org/wiki/Application_programming_interface>

[11] Client-server model <https://en.wikipedia.org/wiki/Client%E2%80%93server_model>

[12]World wide web

<https://en.wikipedia.org/wiki/World_Wide_Web>