voicebot-service

Voicebot wrapper master server

Install dependencies

```
$ pip install -r requirements.txt
```

gRPC Service

```
service VoiceBot {
    rpc ListCallCenter(ListCallCenterRequest) returns (ListCallCenterResponse);
    rpc CallToBot (stream VoiceBotRequest) returns (stream VoiceBotResponse);
}
```

ListCallCenter

```
rpc ListCallCenter(ListCallCenterRequest) returns (ListCallCenterResponse);
```

A function allow client to get all available CallCenter on service

CallToBot

```
rpc CallToBot (stream VoiceBotRequest) returns (stream VoiceBotResponse);
```

A bidirectional streaming function between client and call center

Protobuf

```
protos/voicebot.proto
```

ListCallCenterRequest

The top-level message sent by the client for the `ListCallCenter` method.

This is just an empty message because grpc function required it.

ListCallCenterResponse

The message returned to the client by the `ListCallCenter` method.

Fields	Type	Description
call_centers[]	`CallCenter`	The list of available CallCenter

CallCenter

Description of a CallCenter supported by the service.

Fields	Type	Description
code	`string`	Code of the call center (Examples: cc1, cc2, ...)
name	`string`	Name of the call center

VoiceBotRequest

The top-level message sent by the client for the `CallToBot` method.

Fields	Type	Description
voicebot_config	`VoiceBotConfig`	Provides config information for the request. The first `VoiceBotRequest` message must contain only a `voicebot_config` message.
audio_content	`bytes`	The audio data to be recognized. Sequential chunks of audio data are sent in sequential `VoiceBotRequest` messages. The first `VoiceBotRequest` message must not contain `audio_content` data and all subsequent `VoiceBotRequest` messages must contain `audio_content` data.

VoiceBotResponse

The message returned to the client by the `CallToBot` method. A series of zero or more `VoiceBotResponse` messages are streamed back to the client.

Fields	Type	Description
status	`Status`	Status of response message
text	`string`	The answer text from call center's voicebot
audio_content	`bytes`	The binary of audio data that Voicebot speak the answer text Audio is encoded with WAV format

VoiceBotConfig

The configuration of the call center.

Fields	Type	Description
call_center_code	`string`	Code of the destination call center that client want to connect
is_male	`bool`	Gender of virtual agent If **True** then select male voice If **False** then select female voice

Status

The message with status code and the notify message. It can be either success or error.

Fields	Type	Description
code	`int32`	Status code If **SUCCESS**, then code = 200 If **ERROR**, then code = -1
message	`string`	The notify message

Client Examples

The client example is wrote in Python, take the audio from microphone directly and stream to server, get the response back from server

```
python3 test_client.py
```

The output result should be look like

```
2019-07-10 23:30:44,408 INFO    Connected to call center
.................
2019-07-10 23:30:49,086 INFO    xin chào bạn
........................
2019-07-10 23:30:55,366 INFO    bạn có thể giúp gì cho tôi
...
```