ECM2429 Systems Development 2 Assignment documentation

Release 0.1

Anonymous

Mar 24, 2022

Table of Contents

1 MyGUI 1
1.1 Control.py - The main file. 1

2 IO 3

3 Azure 5

4 Wolfram 7

5 Database 9

6 Conf.py 11

7 Test file for control.py 13

8 Test file for record_play.py 15

9 Test file for azure_logic.py 17

10 Test file for wolfram_logic.py 19

11 Test file for sql_logic.py 21

Python Module Index 23

Index 25

Chapter 1

MyGUI

1.1 Control.py - The main file.

This file holds the UI, and acts as the control unit of the program. It handles the program's main operations by calling other classes and methods and it figures out what is needed to be done with the data that is returned

Use it like this:

```
from main import MyGUI
ui = MyGUI("Virtual Assistant Simulation")
ui.run()
```

class control.**MyGUI** (*name: str*)

This class holds the gui and the code for the main program

__init__ (*name: str*) → None

MyGUI constructor :param name: The name of this thing :type name: str

__weakref__

list of weak references to the object (if defined)

answer_mode (*command*)

This method handles the answer mode for the simulation by calling wolfram api via wolfram.py

click ( )

Called when button clicked

listening_mode ( )

This method handles the listening mode for the Similation

on_off_mode ( )

This method handles the On/Off mode for the Similation

play_mode (*command*)

This method handles the play mode for the simulation by calling sql_logic.py

run ( )

This method starts the tkinter mainloop

IO

record_play.py holds all the logic needed to record audio from a user's device and also play audio The IO class stands for Input/Ouput

class record_play.**IO**

The IO class handles the recording and playing of audio

play (*audio_in: any*)

audio_in is either a path to a WAV file or bytes containing WAV audio.

record (*filename: str, duration=5.0*) → bool

The record method records audio for 5 seconds by using the pyaudio library

Chapter 3

Azure

A class that handles all the interactions with the Azure API

class azure_logic.**Azure**

Azure contains a method that converts a sound/command) to text. The class has another method that turns text to sound

speech_to_text (*sound_as_binary*)

Sends the API request to convert speech to text. It expects sound as a wav file and returns the response object, which contains the text in: response[“content”][“DisplayText”].

text_to_speech (*message*)

This method handles the api call to azure for text to speech

Wolfram

A class that handles all the interactions with the wolfram API

class wolfram_logic.**Wolfram**

Wolfram class contains a constructor and a method to call the wolfram API

call_wolfram (*query*)

This method makes the api call and performs error handling.

Chapter 5

Database

This module handles all the interaction with the Database

class sql_logic.**Database**

Class to handle SQLite operations like playing from a db

play_from_db (*command*) → str

This method runs during play mode. It selects the correct sound to run based on the command from user

Chapter 6

Conf.py

This file stores config variables like API keys

Chapter 7

Test file for control.py

test_control.**test_tkinter** (*mock_tk: unittest.mock.MagicMock, mock_thread*)
Testing that the tkinter module is called

Chapter 8

Test file for record_play.py

test_record_play.**test_play** (*mock_audio: unittest.mock.MagicMock, mock_wave: unittest.mock.MagicMock*)

Testing the play audio method

test_record_play.**test_record** (*mock_audio: unittest.mock.MagicMock, mock_wave: unittest.mock.MagicMock*)

Testing to ensure the record audio method is called

Chapter 9

Test file for azure_logic.py

test_azure_logic.**test_stt** (*mock_req: unittest.mock.MagicMock*)
Testing to ensure that the Speech to Text API is called

test_azure_logic.**test_tts** (*mock_req: unittest.mock.MagicMock*)
Testing to ensure that the Text to Speecjh API is called

Test file for wolfram_logic.py

test_wolfram_logic.**test_call_wolfram** (*mock_get: unittest.mock.MagicMock*)
Testing to ensure the wolfram API is called

Test file for sql_logic.py

test_sql_logic.**test_play_from_db** (*mock_db: unittest.mock.MagicMock*)
Testing to make sure the sqlite module is called

- genindex
- modindex
- search

a

azure_logic, 3

c

conf, 9
control, 1

r

record_play, 1

s

sql_logic, 7

t

test_azure_logic, 15
test_control, 11
test_record_play, 13
test_sql_logic, 19
test_wolfram_logic, 17

w

wolfram_logic, 5

Symbols

__init__() (control.MyGUI method), 1
__weakref__ (control.MyGUI attribute), 1

A

answer_mode() (control.MyGUI method), 1
Azure (class in azure_logic), 5
azure_logic
    module, 3

C

call_wolfram() (wolfram_logic.Wolfram method), 7
click() (control.MyGUI method), 1
conf
    module, 9
control
    module, 1

D

Database (class in sql_logic), 9

I

IO (class in record_play), 3

L

listening_mode() (control.MyGUI method), 1

M

module
    azure_logic, 3
    conf, 9
    control, 1
    record_play, 1
    sql_logic, 7
    test_azure_logic, 15
    test_control, 11
    test_record_play, 13
    test_sql_logic, 19
    test_wolfram_logic, 17
    wolfram_logic, 5
MyGUI (class in control), 1

O

on_off_mode() (control.MyGUI method), 1

P

play() (record_play.IO method), 3
play_from_db() (sql_logic.Database method), 9
play_mode() (control.MyGUI method), 1

R

record() (record_play.IO method), 3
record_play
    module, 1
run() (control.MyGUI method), 1

S

speech_to_text() (azure_logic.Azure method), 5
sql_logic
    module, 7

T

test_azure_logic
    module, 15
test_call_wolfram() (in module test_wolfram_logic), 19
test_control
    module, 11
test_play() (in module test_record_play), 15
test_play_from_db() (in module test_sql_logic), 21
test_record() (in module test_record_play), 15
test_record_play
    module, 13
test_sql_logic
    module, 19
test_stt() (in module test_azure_logic), 17
test_tkinter() (in module test_control), 13
test_tts() (in module test_azure_logic), 17

test_wolfram_logic
    module, 17
text_to_speech() (azure_logic.Azure method), 5

W

Wolfram (class in wolfram_logic), 7
wolfram_logic
    module, 5