

# AmazonEcho documentation

*Release 0.1.0*

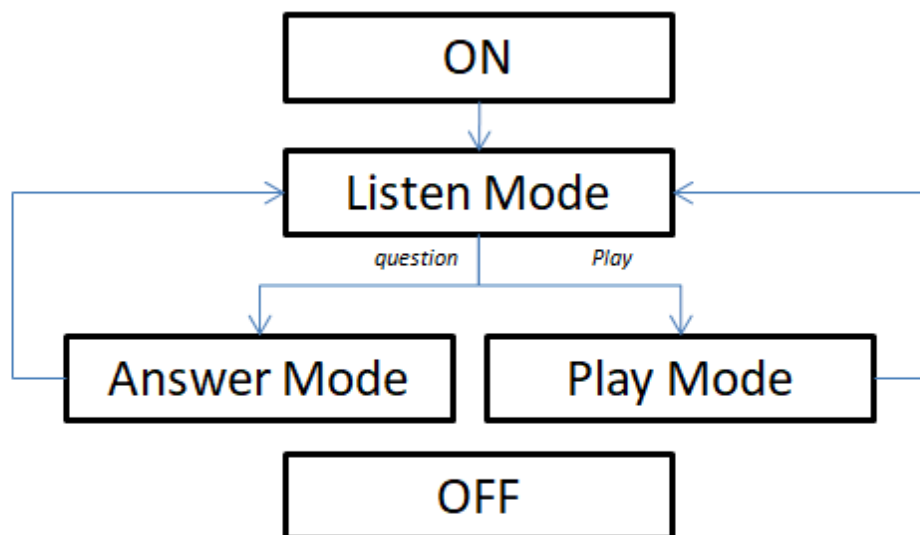
**690065847**

Mar 12, 2022



<b>1</b>	<b>Text to Speech</b>	<b>3</b>
1.1	Text to speech method . . . . .	3
<b>2</b>	<b>Answer Question Mode</b>	<b>5</b>
2.1	Answer question method . . . . .	5
<b>3</b>	<b>Database Configuration</b>	<b>7</b>
3.1	Add data to database . . . . .	7
<b>4</b>	<b>Database Configuration</b>	<b>9</b>
4.1	Retrive data from database . . . . .	9
<b>5</b>	<b>Listening service</b>	<b>11</b>
5.1	Listens & Translate users voice into text . . . . .	11
<b>6</b>	<b>Play service</b>	<b>13</b>
6.1	Plays Binary . . . . .	13
<b>7</b>	<b>GUI &amp; Controller</b>	<b>15</b>
7.1	Runs Amazon Echo. . . . .	15
	<b>Python Module Index</b>	<b>17</b>
	<b>Index</b>	<b>19</b>





**alt** High level overview of Amazon echo program



---

## Text to Speech

---

### 1.1 Text to speech method

#### 1.1.1 Text to speech converter

Use it like this:

```
from txt_to_speech import text_to_speech  
  
text_to_speech("what is your name")
```

`txt_to_speech.text_to_speech ( text: str ) → str`

function converts text to speech uses the request module to send text content to microsoft returns binary data.

**Parameters text** – A string of text.





---

## Answer Question Mode

---

### 2.1 Answer question method

---

*Use it like this::*

```
from answer_service import answer_mode question = "who is the queen of england" answer_service(question)
```

`answer_service.answer_mode ( question: str ) → str`

Takes a users question in string format Returns results from wolframalpha api in json format  
raises Error when question is not found or server error



---

## Database Configuration

---

### 3.1 Add data to database

#### 3.1.1 Add data to database

*Example of a method use:*

```
from database_entry import insert_file
insert_file('quiet-music.wav', 'sample.db', 'audio', "music" , "quiet")
```

`database_entry.convert_into_binary (file_path: str) → str`

Takes a file with wav format converts to binary

`database_entry.create_db_table (db_name: str) → str`

db\_name creates the database and schema in sqlite

`database_entry.insert_file ( file_name: str, db_name: str, table_name: str, category: str, SHORT_DESC: str)`

inserts audio data into sqlite and stores it as a blob format.

`database_entry.sqlite_connect (db_name: str) → str`

takes database name and attempts to connect



---

## Database Configuration

---

### 4.1 Retrive data from database

#### 4.1.1 Retrive data from database

Use it like this:

```
from database_retriv import retrieve_file  
retrieve_file("dog barking")
```

database\_retriv.**retrieve\_file** (*file\_name: str*) → str

takes in a string and searches the string value in the database when found BLOB in data format is returned/played else not found error returned



---

## Listening service

---

### 5.1 Listens & Translate users voice into text

#### 5.1.1 Listens & Translate users voice into text

*Use it like this::*

`record()` # this will start voice recording `speech_to_text()` # convert stored audio to text

`listening_service.record ( duration=5.0 )` → bool  
pyaudio records users voice and converts to wav format

`listening_service.speech_to_text ( )` → None  
reads the dd.wav and sends the data to microsoft api to translate into english text





---

## Play service

---

### 6.1 Plays Binary

#### 6.1.1 Plays Binary

*Use it like this:*

```
from play_service import play
play()
```

`play_service.play(audio_in: any)`  
takes in wav data and plays it



---

## GUI & Controller

---

### 7.1 Runs Amazon Echo

#### 7.1.1 Runs Amazon Echo program

`main.Simpletoggle ( )` → None

ON/OFF mode for echo with welcome messages.

`main.controller ( )` → None

this function acts as a directional service for play and questions

`main.create_thread ( )` → None

threading to solve frozen ON button.

- genindex
- modindex
- search



## a

`answer_service`, 3

## d

`database_entry`, 5

`database_retriv`, 7

## l

`listening_service`, 9

## m

`main`, 13

## p

`play_service`, 11

## t

`txt_to_speech`, 1



## A

answer\_mode() (in module answer\_service), 5  
answer\_service  
    module, 3

## C

controller() (in module main), 15  
convert\_into\_binary() (in module database\_entry), 7  
create\_db\_table() (in module database\_entry), 7  
create\_thread() (in module main), 15

## D

database\_entry  
    module, 5  
database\_retriv  
    module, 7

## I

insert\_file() (in module database\_entry), 7

## L

listening\_service  
    module, 9

## M

main  
    module, 13  
module  
    answer\_service, 3  
    database\_entry, 5  
    database\_retriv, 7  
    listening\_service, 9  
    main, 13  
    play\_service, 11  
    txt\_to\_speech, 1

## P

play() (in module play\_service), 13

play\_service  
    module, 11

## R

record() (in module listening\_service), 11  
retrieve\_file() (in module database\_retriv), 9

## S

Simpletoggle() (in module main), 15  
speech\_to\_text() (in module listening\_service),  
    11  
sqlite\_connect() (in module database\_entry), 7

## T

text\_to\_speech() (in module txt\_to\_speech), 3  
txt\_to\_speech  
    module, 1

