

chat.html

new_audio_js/upload.php
входной параметр внутри upload.php называется `$_FILES`
и мы перекладываем с помощью функции `move_uploaded_file` файл на постоянное хранение на сервере `as wav files` (хранилище голосов надо будет периодически обрабатывать - с помощью программ демонов - CRON *the most famous daemon)

с помощью `echo $html;` идет в `e.target.responseText`

2 curl query

text идет в параметр `$html`

3

cgi-bin/audio.py
принимает параметр `audio_file_name` из `upload.php`
мы скормливаем `audio_file_name` в `google speech recognizer`
и он кладет распознанный голос в переменную `text`
А потом этот `text` записываем на сервер в текстовый файл
В конце этот текст командой `print(text)` возвращается в вызывающий скрипт

1. new_audio_js/app.js
`createDownloadLink`
принимает `blob` в качестве параметра и отправляет его на сервер аякс запросом с помощью механизма `new FormData()`

внутри конструкции приходит ответ сервера в параметр `e.target.responseText`

```
xhr.onload=function(e) {  
  if(this.readyState === 4) {  
    console.log("Server returned: ",e.target.responseText);  
    send_voice_text(e.target.responseText);  
  }  
};
```

new_audio_js/recorder.js
(ваш голос собирает в `blob`)

4

5

chat.html
`send_voice_text(param)`
в `param` принимает распознанный текст
1) вставляет в тело браузера(чата), чтобы человек увидел корректность распознавания
2) потом отправляем аякс запрос с распознанным голосом в `cgi-bin/song.py` (чтобы обработать вопрос пользователя и сформировать звуковой ответ сервера)

param1

6

7

имя mp3 файла

cgi-bin/song.py
этот текст будет доставаться из текстового файла
ответ сервера на этот текст будет доставаться из базы данных
создаст `mp3` файл и отдаст в браузер имя этого файла

База данных