Откомментируйте код прямо в тексте:

**#Importing libraries**

**#Импорт библиотек**

import xml.etree.ElementTree as XmlElementTree

import httplib2

import uuid

from config import \*\*\*

\*\*\*\_HOST = '\*\*\*'

\*\*\*\_PATH = '/\*\*\*\_xml'

CHUNK\_SIZE = 1024 \*\* 2

def speech\_to\_text(filename=None, bytes=None, request\_id=uuid.uuid4().hex, topic='notes', lang='ru-RU',

key=\*\*\*\_API\_KEY):

**#Open the file and read it if no bytes throw an exception**

**#Открываем файл если нет байтов кидаем исключение**

if filename:

with open(filename, 'br') as file:

bytes = file.read()

if not bytes:

raise Exception('Neither file name nor bytes provided.')

**#converting to the special format**

**#конвертировать в специальный формат**

bytes = convert\_to\_pcm16b16000r(in\_bytes=bytes)

**#Creating a request to the API**

**#Создаем запрос к API**

url = \*\*\*\_PATH + '?uuid=%s&key=%s&topic=%s&lang=%s' % (

request\_id,

key,

topic,

lang

)

**#reading a block of bytes**

**#читаем блок размера байтов**

chunks = read\_chunks(CHUNK\_SIZE, bytes)

**# Setting the connection with the the HOST**

**# устанавливаем соединение с HOST**

connection = httplib2.HTTPConnectionWithTimeout(\*\*\*\_HOST)

**# connect to the server specified previous**

**# подключаемся к серверу отмеченный ранее**

connection.connect()

**# the following line sends request to the server**

**# отправляем request к серверу**

connection.putrequest('POST', url)

**#sends header to the server**

**#посылаем заголовок к серверу**

connection.putheader('Transfer-Encoding', 'chunked')

connection.putheader('Content-Type', 'audio/x-pcm;bit=16;rate=16000')

**#sends blank line telling that this is the end of headers**

**#отправляем пустой код который обозначает окончание заголовок**

connection.endheaders()

**#loop through the chunks created above and sending the connection**

**#лупим через chunks который был создан ранее и отправляем соединение**

for chunk in chunks:

connection.send(('%s\r\n' % hex(len(chunk))[2:]).encode())

connection.send(chunk)

connection.send('\r\n'.encode())

connection.send('0\r\n\r\n'.encode())

**#getting response**

**#получаем response**

response = connection.getresponse()

**#Checking the response status**

**#Получаем респонс и обрабатываем**

if response.code == 200:

response\_text = response.read()

**#converting response\_text to the xml and storing it in the variable called “xml”**

**#конвертируем response\_text на xml и храним данные в переменном(variable) который называется “xml”**

xml = XmlElementTree.fromstring(response\_text)

**#checking if xml attribute’s success int equals to 1**

**#if it is then convert it to the float element**

**# проверяем статус атрибута если равна к 1 то конвертируем к float элемент**

if int(xml.attrib['success']) == 1:

max\_confidence = - float("inf")

text = ''

**#loop through each child element in xml**

**#лупим xml**

for child in xml:

if float(child.attrib['confidence']) > max\_confidence:

text = child.text

max\_confidence = float(child.attrib['confidence'])

if max\_confidence != - float("inf"):

return text

else:

raise SpeechException('No text found.\n\nResponse:\n%s' % (response\_text))

else:

raise SpeechException('No text found.\n\nResponse:\n%s' % (response\_text))

else:

raise SpeechException('Unknown error.\nCode: %s\n\n%s' % (response.code, response.read()))

**#creating an exception**

**#создаем исключение**

сlass SpeechException(Exception):

pass