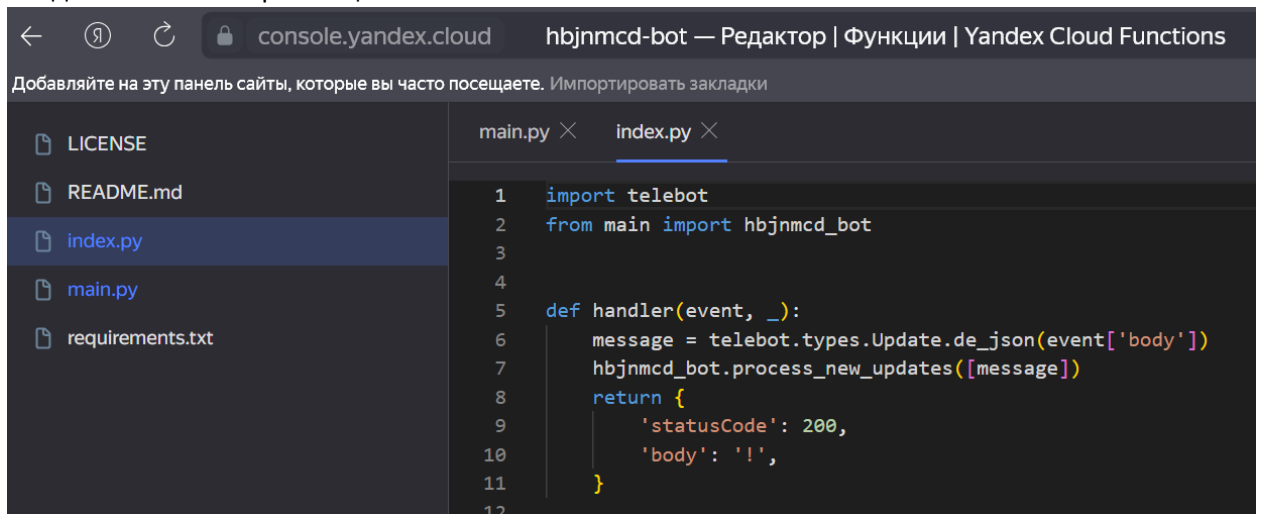


Telegram-бот

Создала Нечаева Наталья

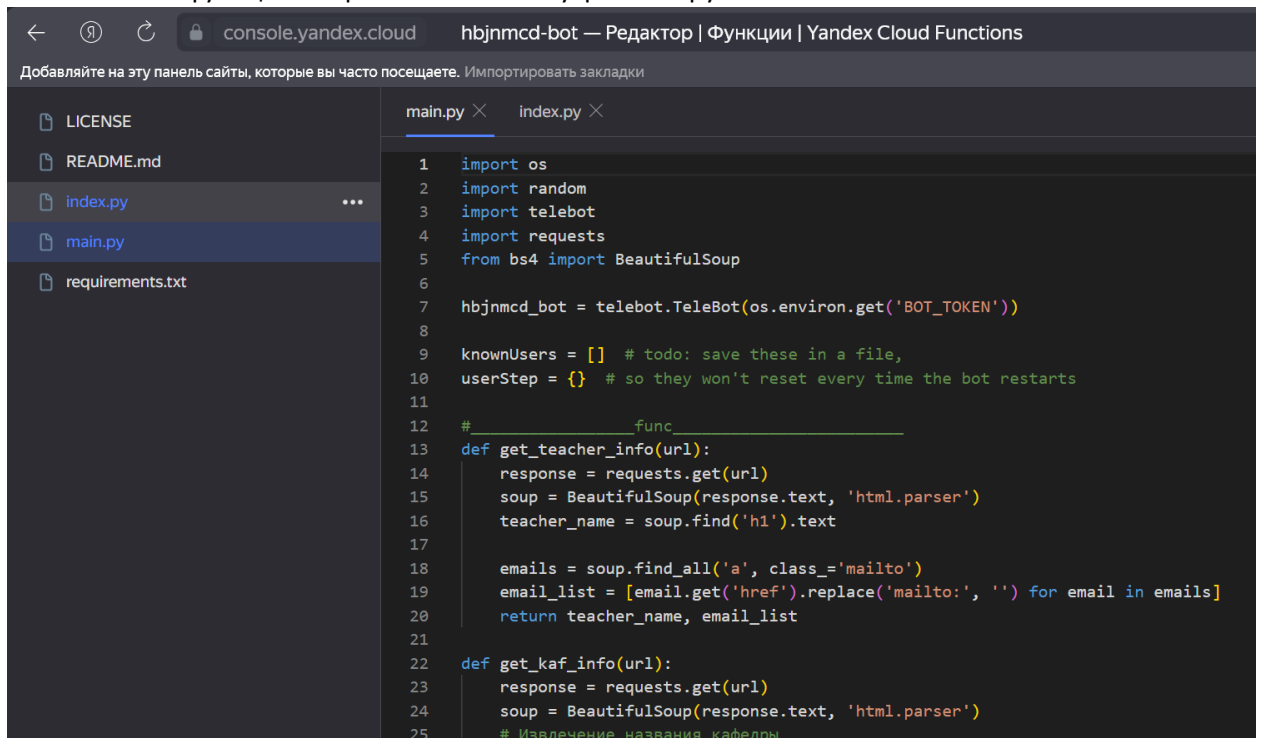
Созданный бот был размещен на Yandex.Cloud Functions:



The screenshot shows the Yandex Cloud Functions console interface. The browser address bar displays 'console.yandex.cloud' and the page title is 'hbjnmcd-bot — Редактор | Функции | Yandex Cloud Functions'. A sidebar on the left lists files: LICENSE, README.md, index.py (selected), main.py, and requirements.txt. The main area shows the code for index.py:

```
1 import telebot
2 from main import hbjnmcd_bot
3
4
5 def handler(event, _):
6     message = telebot.types.Update.de_json(event['body'])
7     hbjnmcd_bot.process_new_updates([message])
8     return {
9         'statusCode': 200,
10        'body': '!',
11    }
12
```

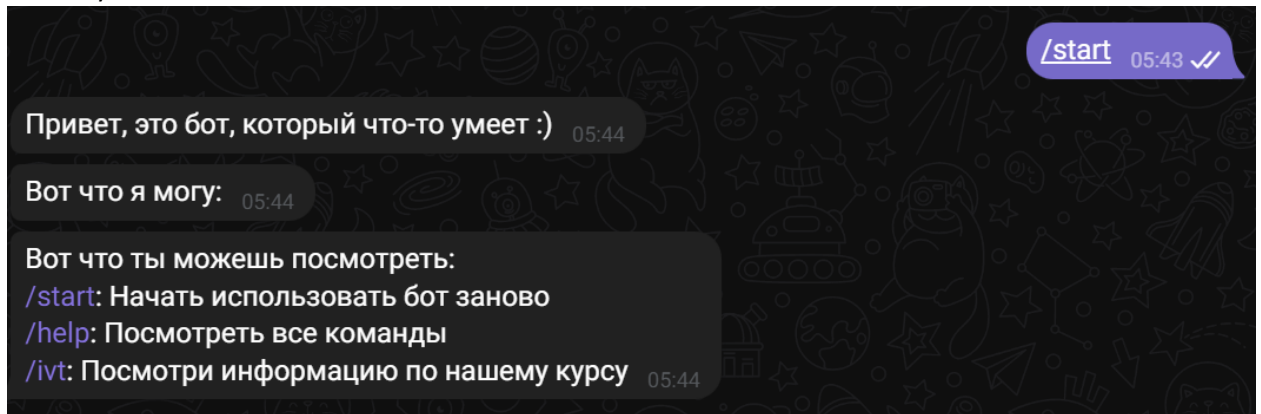
Основной его функционал располагается внутри main.py:



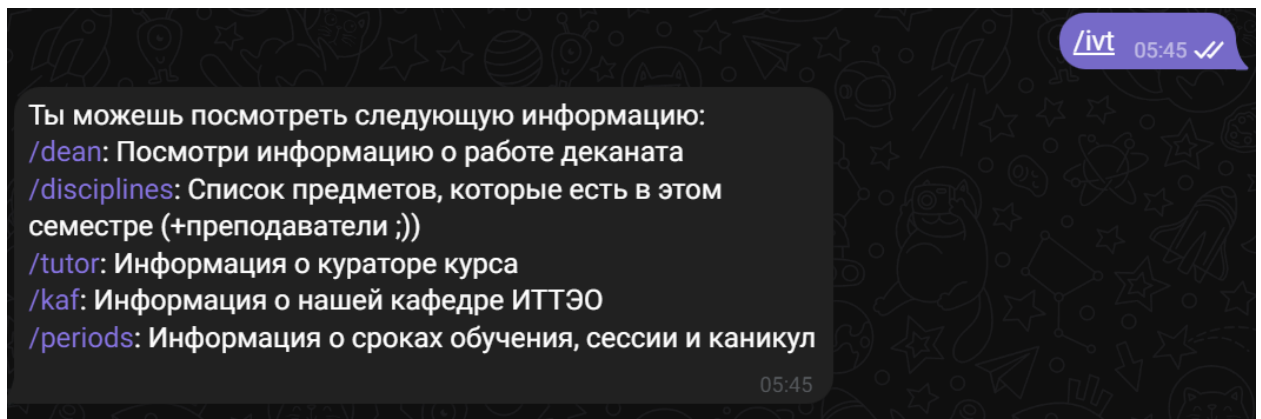
The screenshot shows the Yandex Cloud Functions console interface with the main.py file selected. The browser address bar displays 'console.yandex.cloud' and the page title is 'hbjnmcd-bot — Редактор | Функции | Yandex Cloud Functions'. The sidebar on the left lists files: LICENSE, README.md, index.py, main.py (selected), and requirements.txt. The main area shows the code for main.py:

```
1 import os
2 import random
3 import telebot
4 import requests
5 from bs4 import BeautifulSoup
6
7 hbjnmcd_bot = telebot.TeleBot(os.environ.get('BOT_TOKEN'))
8
9 knownUsers = [] # todo: save these in a file,
10 userStep = {} # so they won't reset every time the bot restarts
11
12 # _____func_____
13 def get_teacher_info(url):
14     response = requests.get(url)
15     soup = BeautifulSoup(response.text, 'html.parser')
16     teacher_name = soup.find('h1').text
17
18     emails = soup.find_all('a', class_='mailto')
19     email_list = [email.get('href').replace('mailto:', '') for email in emails]
20     return teacher_name, email_list
21
22 def get_kaf_info(url):
23     response = requests.get(url)
24     soup = BeautifulSoup(response.text, 'html.parser')
25     # Извлечение названия кафедры
```

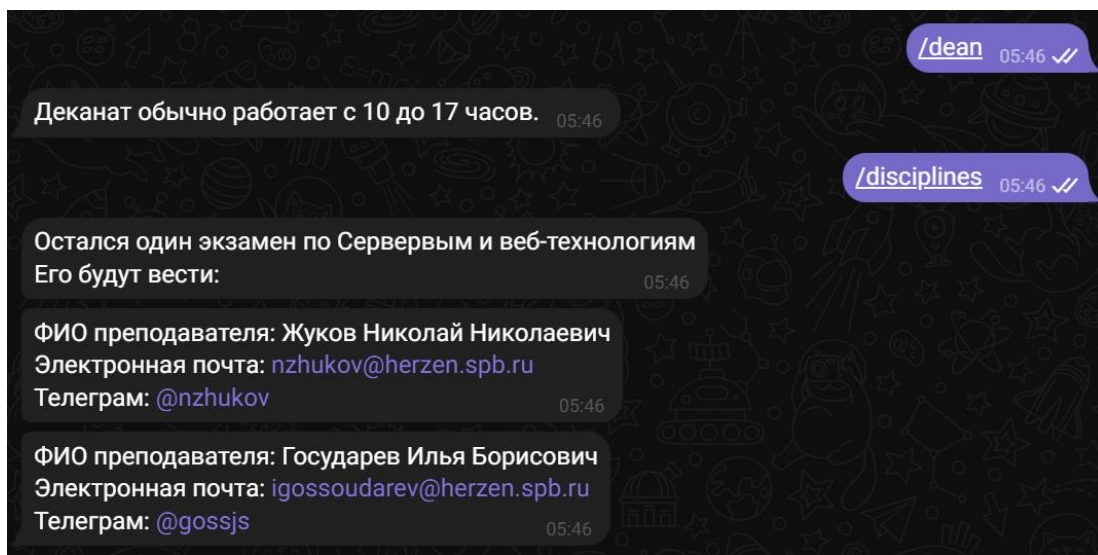
Что он умеет:



Как можно увидеть, при запуске он показывает все доступные на данный момент вызовы (их немного).

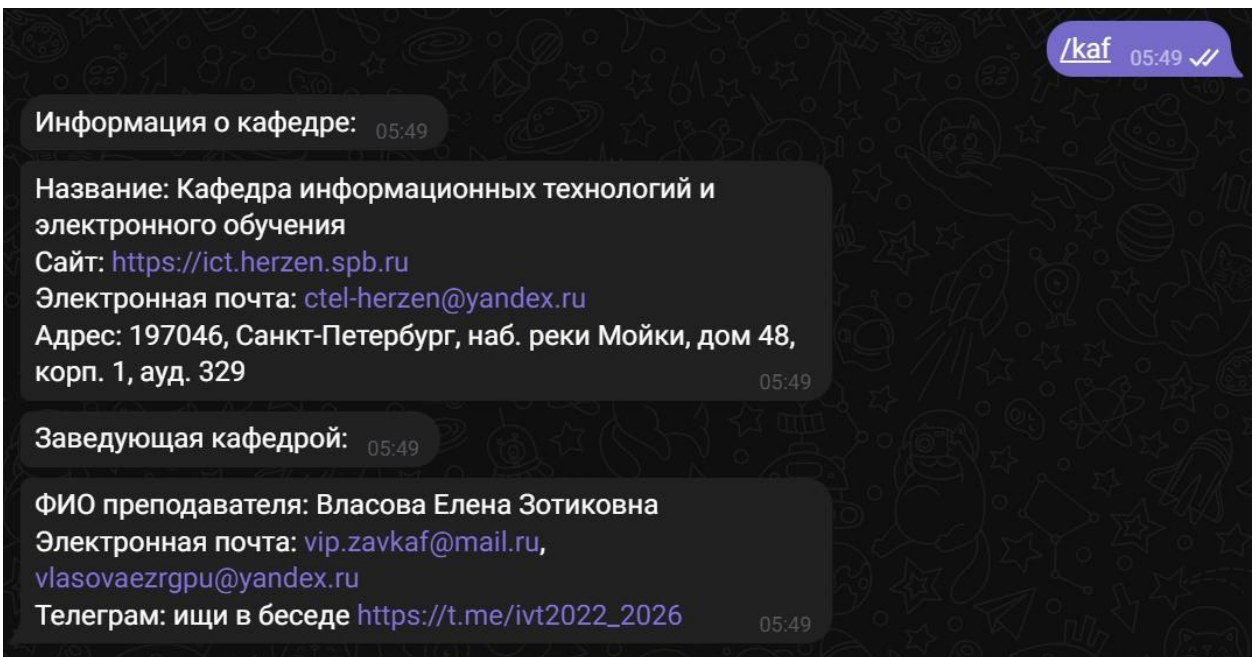


Этот бот рассчитан на студентов, поэтому обращение к пользователю неформальное.



В дальнейшем я планирую выводить всю информацию о том, какие дисциплины ведутся в данный момент на курсе и кто их ведет. Информация о преподавателях берется отсюда:

<https://ict.herzen.spb.ru/> (со страниц преподавателей; только телеграмм я вводила самостоятельно)



Фрагменты кода, которые отвечают за данные с других сайтов:

```
import requests
from bs4 import BeautifulSoup
def get_teacher_info(url):

    response = requests.get(url)
    soup = BeautifulSoup(response.text, 'html.parser')
    teacher_name = soup.find('h1').text

    emails = soup.find_all('a', class_='mailto')
    email_list = [email.get('href').replace('mailto:', '') for email in emails]
    return teacher_name, email_list

def get_kaf_info(url):
    response = requests.get(url)
    soup = BeautifulSoup(response.text, 'html.parser')
    # Извлечение названия кафедры
    department_name_tag = soup.find('div', class_='title-h2')
    department_name = department_name_tag.h2.text if department_name_tag else
'Название не найдено'
    # Извлечение сайта кафедры
    department_site_tag = soup.find(
        lambda tag: tag.name == 'a' and 'href' in tag.attrs and
'ict.herzen.spb.ru' in tag['href'])
    department_site = department_site_tag['href'] if department_site_tag else
'Сайт не найден'
    # Извлечение email кафедры
    department_email_tag = soup.find(
        lambda tag: tag.name == 'a' and 'href' in tag.attrs and 'mailto:' in
tag['href'])
    department_email = department_email_tag['href'].replace('mailto:',
'' ) \
```

```
        if department_email_tag else 'Email не найден'
    # Извлечение адреса кафедры
    department_address_tag = soup.find(
        lambda tag: tag.name == 'a' and 'href' in tag.attrs and 'maps' in
tag['href'])
    department_address = department_address_tag.text if department_address_tag
else 'Адрес не найден'

    return department_name, department_site, department_email, department_address
```