

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С. П. КОРОЛЕВА»  
(САМАРСКИЙ УНИВЕРСИТЕТ)

ФАКУЛЬТЕТ ИНФОРМАТИКИ  
КАФЕДРА ПРОГРАММНЫХ СИСТЕМ

**Отчёт по дополнительному заданию**

Разработка программы, способной добавить нового пользователя в ТЛН,  
запустить его сессию, и загрузить в его каталог текстовый файл

Выполнил:  
Кейн Р. Д.  
гр.6303-090301D  
Хитёт В.В.  
гр.6304-090301D

Проверил:  
Востокин С. В.

Самара 2023

## ВВЕДЕНИЕ

Для выполнения данного задания:

- 1) Был взят образ операционной системы Ubuntu 20.04 LTS для установки на виртуальную машину VirtualBox  
(<https://releases.ubuntu.com/focal/>)
- 2) На виртуальную машину VirtualBox установлен клиент сервера TLJH  
(<https://tljh.jupyter.org/en/latest>)
- 3) Было проведено ознакомление с документацией JupyterHub REST Api  
(<https://jupyterhub.readthedocs.io/en/stable/reference/rest-api.html#/>)

## НАПИСАНИЕ ПРОГРАММЫ

Напишем программу, которая должна выполнять следующие действия:

- 1) Получить имя нового пользователя и попытаться его добавить;
- 2) Запустить сервер нового пользователя и получить токен для работы с REST API;
- 3) Получить название файла и его содержимое;
- 4) Загрузить файл в каталог пользователя;
- 5) Завершить работу сервера пользователя.

Листинг программы:

```
import requests
import json

api_url = 'http://10.0.2.15/hub/api' # base api url
token = "048f316054e04903abd258a9429e37a9" # admin token

flag = True

while flag:
    new_user_name = input("Введите имя нового пользователя: ")
    #new_user_name = "new_user"

    new_user_data = {
        'name': new_user_name,
        'admin': False,
        'server': False,
        'kind': 'user'
    }

    headers = {
        'Authorization': f'token{token}',
        'Content-Type': 'application/json'
    }
    add_user_response = requests.post(api_url + '/users/' + new_user_name,
headers={'Authorization': f'token {token}'}) # request to add new use
    if add_user_response.status_code == 201:
        print("Пользователь успешно создан")
        flag = False
    else:
        print("Пользователь с таким именем уже существует")
        answer = input("Желаете продолжить от имени старого пользователя?
y/n: ")
        if answer == "y": flag = False

start_server_response = requests.post(api_url + '/users/' + new_user_name +
'/server', headers={'Authorization': f'token {token}'}) # starting new user
server

new_token = ''
get_token_response = requests.post(api_url + '/users/' + new_user_name +
```

```

'/tokens', headers={'Authorization': f'token {token}'}, json={'username':
new_user_name, 'password': 'password'})
if get_token_response.status_code == 201:
    new_token = get_token_response.json()['token']
    print("Токен успешно получен: " + new_token)

file_name = input("Введите имя файла: ")
#file_name = "testfile"
fdat = input("Введите содержимое файла: ")
#fdat = "testdata111"

data = {
    'content': fdat,
    'format': 'text'
}

create_file_response = requests.put('http://10.0.2.15/user/'+new_user_name+'
/api/contents/' + file_name, json=data, headers={'Authorization': 'token ' +
new_token})

if create_file_response.status_code != 201:
    print("Файл с таким именем уже существует, он будет пересоздан!")
    delete_file_response =
requests.delete('http://10.0.2.15/user/'+new_user_name+'/api/contents/'+file_
name, headers={'Authorization': 'token ' + new_token})
    create_file_response = requests.put('http://10.0.2.15/user/' +
new_user_name + '/api/contents/' + file_name, json=data,
headers={'Authorization': 'token ' + new_token})
else:
    print("Файл успешно добавлен")
wait_press = input("Введите что-нибудь, чтобы остановить сервер")

stop_server_response = requests.delete(api_url + '/users/' + new_user_name +
'/server', headers={'Authorization': f'token {new_token}'}) # starting new
user server

print("Сервер завершил работу")

```

Для работы программы необходимо заменить значение переменной «token» на значение токена, созданного для вашего пользователя с правами администратора, а так же заменить значение переменной «api\_url» на адрес, по которому у вас находится сервер JupyterHub.

## ТЕСТИРОВАНИЕ

Результаты работы представлены на рисунках 1-5

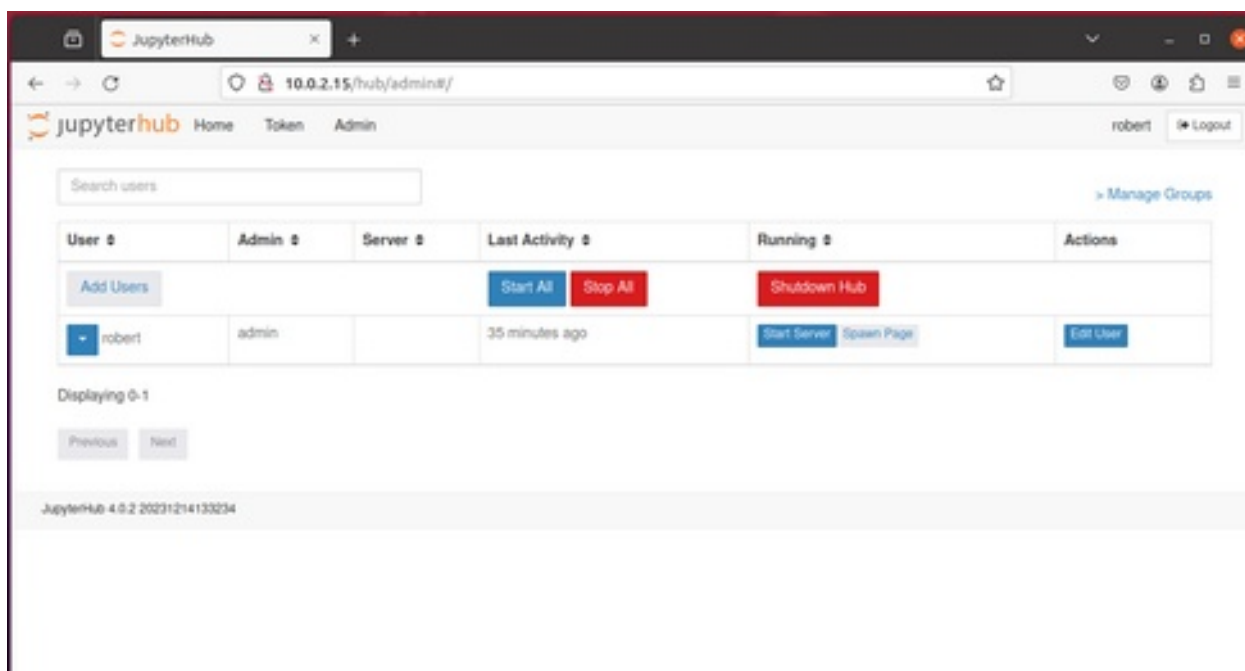


Рисунок 1 – список пользователей JupyterHub до запуска программы



Рисунок 2 – демонстрация работы программы, добавление нового пользователя, запуск его сервера, получение токена для работы с REST API, получение названия и содержимого файла, загрузка файла в каталог

пользователя

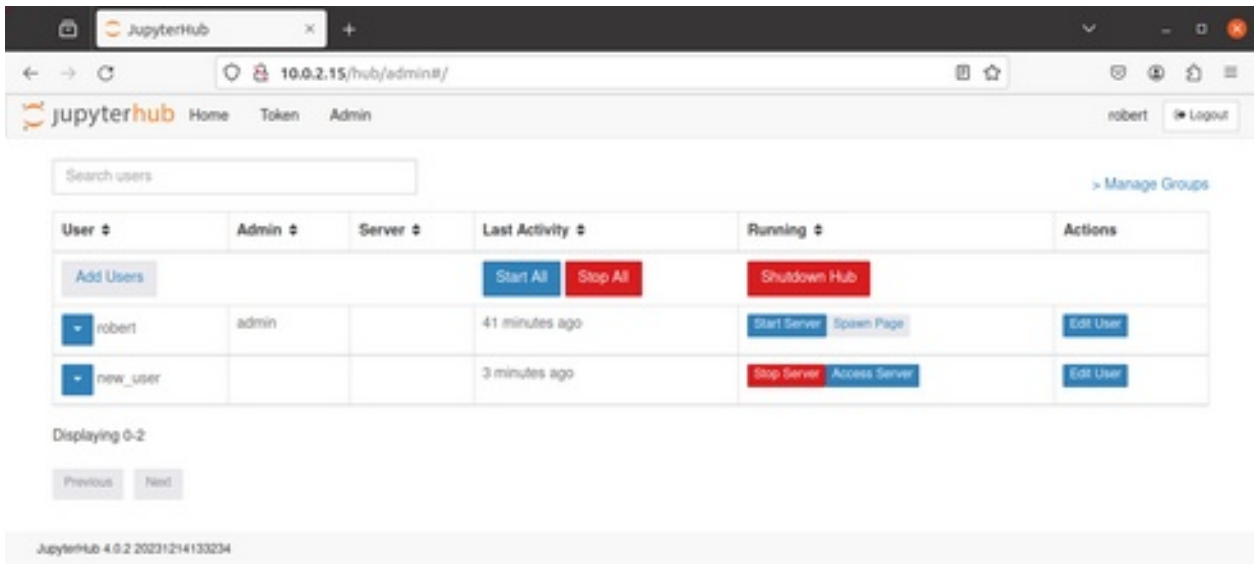


Рисунок 3 – список пользователей JupyterHub после выполнения программы

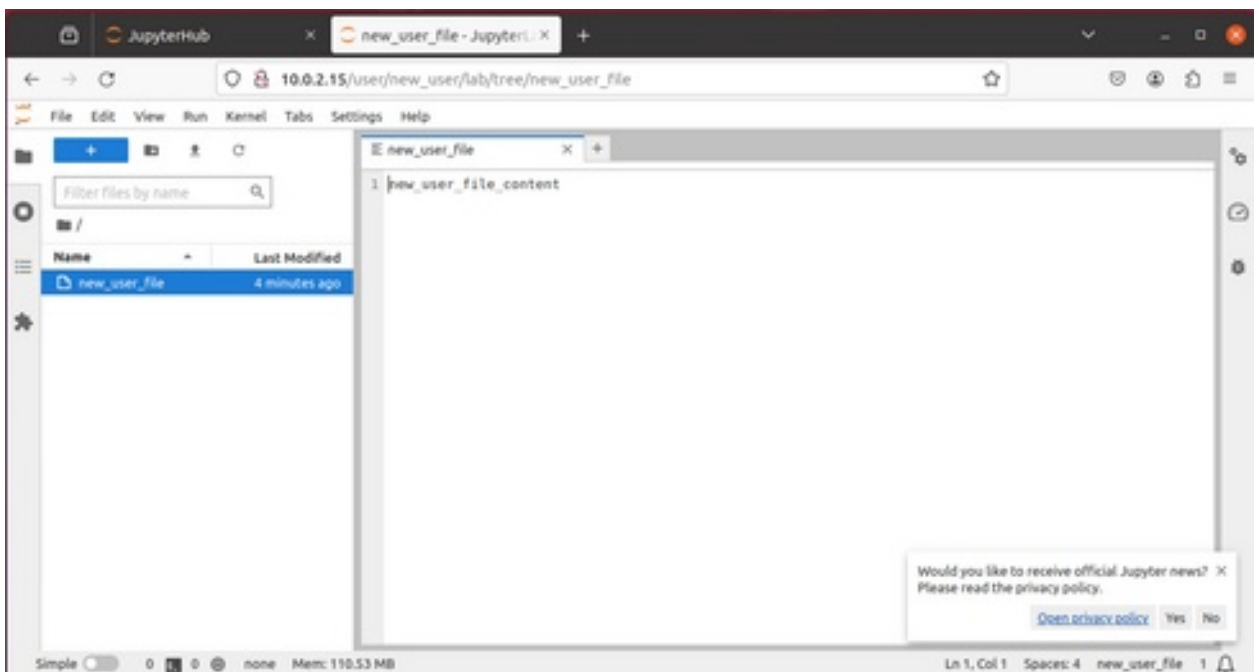


Рисунок 4 – демонстрация каталога нового пользователя, а так же содержимого загруженного файла

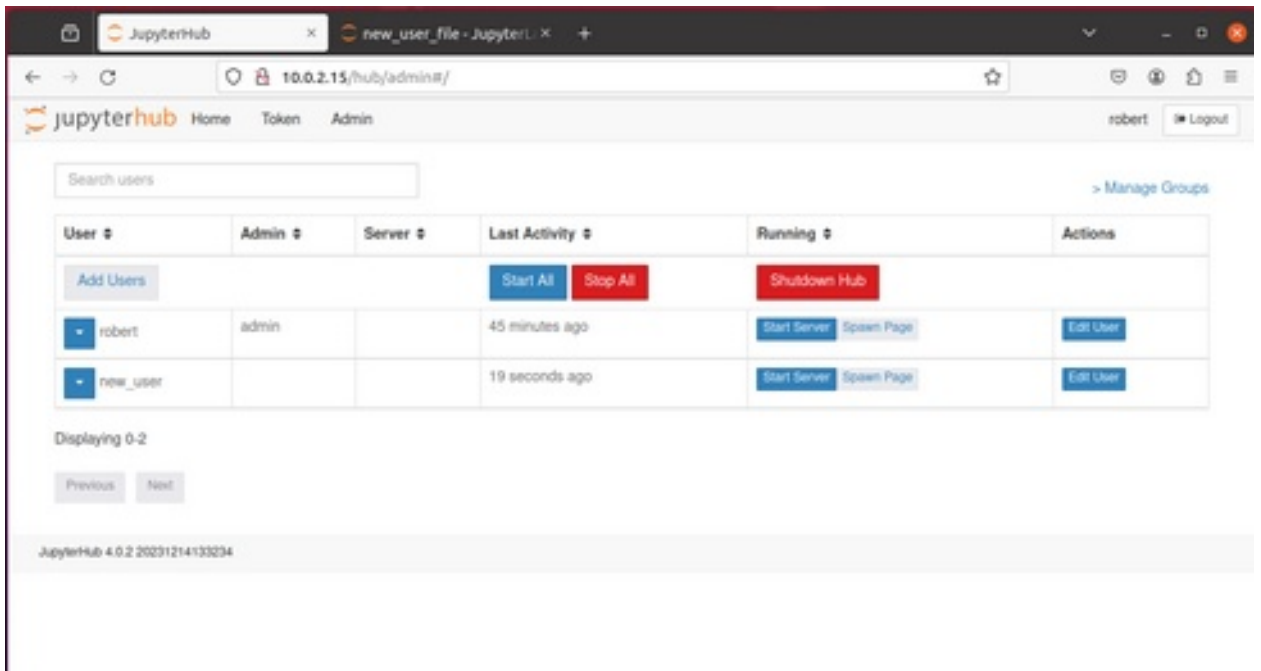


Рисунок 5 – в конце выполнения программы, сервер нового пользователя выключается