

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное
учреждение высшего образования

“Пермский государственный национальный исследовательский
университет”

Лабораторная работа №1

*по дисциплине «Технологии разработки распределенных
приложений»*

Работу выполнила студентка
группы КМБ-16 5 курса
механико-математического
факультета

Кузнецова Александра
Дмитриевна

«2» октября 2020 г.

СОДЕРЖАНИЕ

Постановка задачи	3
Технологии и структура приложения	3
Описание приложения	4
Соответствие поставленной задаче	12
Список источников	13

Постановка задачи

Цель: Изучение возможностей использования API современных популярных информационных ресурсов внешними приложениями.

Формируемая компетенция: способность применять на практике теоретические основы и общие принципы разработки распределенных систем.

Организация выполнения работы: Каждый студент выполняет индивидуальное задание.

Требования к выполнению работы:

1. Разработанное приложение должно предоставлять возможности создания, чтения, обновления и удаления некоторых объектов выбранного информационного ресурса (выполнения CRUD-операций).
2. CRUD-операции должны выполняться при помощи API выбранного ресурса, причём по крайней мере один тип операции должен требовать аутентификации на нём.
3. Должна быть возможность выполнения по крайней мере двух CRUD-операций для одного и того же объекта выбранного информационного ресурса.

Технологии и структура приложения

В качестве информационного ресурса был выбран GitHub [1]. GitHub — хостинг IT-проектов, содержит в открытом доступе множество репозиторий, содержащих файлы с кодом, документацией и т. д. в соответствующих форматах.

Приложение совместимо с семейством ОС Linux. Технологии:

- Язык - python 3.7.4 [2];
- Среда разработки - PyCharm Professional 2019.3.3 [3], предоставленная по индивидуальной студенческой лицензии JetBrains [4].
- Для взаимодействия с API GitHub была использована библиотека PyGitHub 1.53 [5];
- Другие библиотеки, используемые в приложении:
 - PyYAML 5.3.1 [6];
 - cryptography 3.1.1 [7];
 - click 7.1.2 [8];

Итоговое приложение составное, содержит следующие файлы:

- cmd.py - содержит определения параметров командной строки и логику выполнения команд с параметрами;
- githubapi.py - содержит функции для работы с GitHub;
- githubapi_config.yaml - конфигурационный файл;

В конфигурационном файле githubapi_config.yaml задается местоположение двух файлов, порождаемых приложением:

- log_file - файл для журналирования событий приложения;
- cred_file - файл для сохранения параметров последней успешно запущенной сессии;

Описание работы приложения

В информационном источнике были выделены две основные сущности - репозиторий и файл. Для каждой из сущностей были реализованы CRUD-операции.

Список всех флагов и их описания доступны по опции `--help`:

```
(venv) (base) 21:57:42 [cuprumtan@localhost github_API_test]$ python3 cmd.py --help
Usage: cmd.py [OPTIONS]

Options:
  -u, --username TEXT      GitHub username
  -p, --password TEXT      GitHub password
  -L, --repolist            Show current user repositories
  --privacy                Show repositories privacy
  --languages              Show repositories languages
  -C, --createrepo TEXT    Create new repository
  -D, --deleterepo TEXT    Delete new repository
  -T, --repocontents TEXT  Show repository contents
  --recursively            Show repository contents recursively
  -R, --renamerepo TEXT    Rename repository. Usage: -R REPO,NEW_NAME
  -P, --repoprivacy TEXT   Edit repository privacy. Usage: -P REPO,PRIVACY
                           where PRIVACY = [T|F]. T for private, F for public

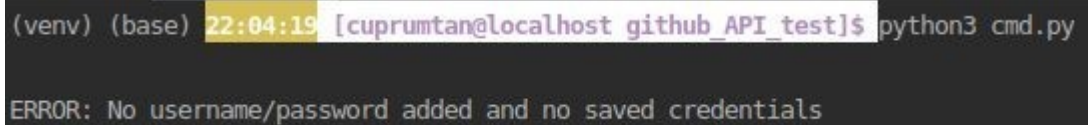
  -F, --findinrepo TEXT    Find files in repo by type. Usage: -F REPO,TYPE
  --repo TEXT              Repository name
  -c, --createfile TEXT    Create new file in repository
  --content TEXT           File content. Usage: --content "WRITE CONTENT HERE"
  -l, --loadfile TEXT      Load new file in repository
  --path TEXT              File path
  -d, --deletefile TEXT    Delete file from repository
  --help                  Show this message and exit.
```

Рисунок 1, доступные флаги приложения и их описание

Авторизация

Приложение работает только с авторизованными пользователями. Авторизация происходит с помощью логина и пароля. При каждой успешной попытке авторизации логин и пароль пользователя сохраняются в зашифрованном виде в файл `cred_file`. Можно задать логин и пароль явно с помощью флагов `-u` и `-p` соответственно. Если файл `cred_file` существует и флаги `-u` и `-p` не заданы, информация берется из этого файла, т.е. используются данные последней успешной сессии.

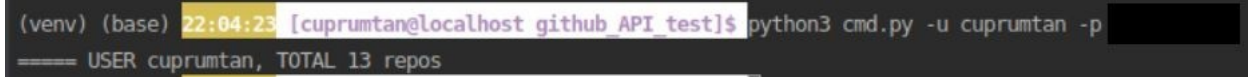
Попытка запуска приложения без флагов `-u` и `-p` и файла `cred_file` (первый запуск приложения):



```
(venv) (base) 22:04:19 [cuprumtan@localhost github_API_test]$ python3 cmd.py
ERROR: No username/password added and no saved credentials
```

Рисунок 2, запуск приложения без флагов `-u` и `-p` и файла `cred_file`

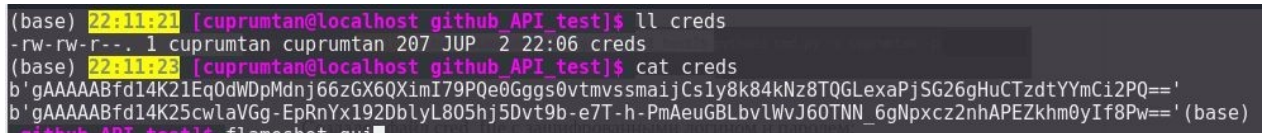
Запуск приложения с явно заданными флагами:



```
(venv) (base) 22:04:23 [cuprumtan@localhost github_API_test]$ python3 cmd.py -u cuprumtan -p [REDACTED]
===== USER cuprumtan, TOTAL 13 repos
```

Рисунок 3, запуск приложения с явно заданными флагами

Создался файл `cred_file` с зашифрованными логином и паролем:

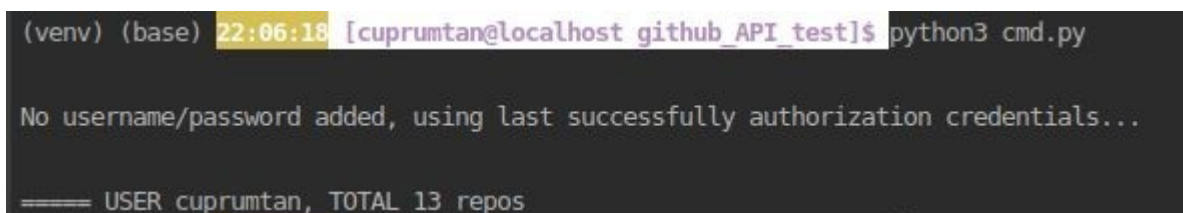


```
(base) 22:11:21 [cuprumtan@localhost github_API_test]$ ll creds
-rw-rw-r-- 1 cuprumtan cuprumtan 207 JUP  2 22:06 creds
(base) 22:11:23 [cuprumtan@localhost github_API_test]$ cat creds
b'gAAAAABfd14K21Eq0dWpMdnj66zGX6QXimI79PQe0Gggs0vtmvssmaijCsly8k84kNz8TQGLexaPjSG26gHuCTzdtYYmCi2PQ=='
b'gAAAAABfd14K25cwLaVgG-EpRnYx192DblyL805hj5Dvt9b-e7T-h-PmAeuGBLbvlWvJ60TNN_6gNpxcz2nhAPEZkhm0yIf8Pw==' (base)
github_API_test: flamechot, guid
```

Рисунок 4, содержимое `cred_file`

Шифрование реализовано с помощью конструкции Фернета [9]. Конструкция Фернета предполагает использование шифра AES128 в режиме сцепления блоков с проверкой целостности с помощью SHA256.

С `cred_file` возможен вход без указания параметров:



```
(venv) (base) 22:06:18 [cuprumtan@localhost github_API_test]$ python3 cmd.py
No username/password added, using last successfully authorization credentials...
===== USER cuprumtan, TOTAL 13 repos
```

Рисунок 5, вход без указания параметров

Сразу после успешной аутентификации выполняется одна из CRUD-операций - чтение списка всех репозиториях пользователя и их подсчет. На изображении выше показано, что у текущего пользователя 13 репозиториях.

В приложении доступны операции на уровне репозиториях и на уровне файлов. Флаги для работы с репозиториями обозначены заглавными буквами, для работы с файлами - строчными.

Операции с репозиториями

Получение списка всех репозиториях пользователя (-L) с их статусом приватности (--privacy) и основным языком (--language):

```
(venv) (base) 22:17:35 [cuprumtan@localhost github_API_test]$ python3 cmd.py -L --privacy --languages

No username/password added, using last successfully authorization credentials...

===== USER cuprumtan, TOTAL 13 repos

===== USER cuprumtan REPOSITORIES:
--- asm (PUBLIC)
    main language: Assembly
--- cool-glasses-generator (PRIVATE)
    main language: Jupyter Notebook
--- db_import_export (PUBLIC)
    main language: Java
--- github-pages-old (PRIVATE)
    main language: HTML
```

Рисунок 6, получение списка всех репозиториях пользователя с их статусом приватности и основным языком

Создание репозитория (-C [имя]):

```
(venv) (base) 22:17:48 [cuprumtan@localhost github_API_test]$ python3 cmd.py -C test_repo

No username/password added, using last successfully authorization credentials...

===== USER cuprumtan, TOTAL 13 repos

Successfully created new repo test_repo
```

Рисунок 7, создание репозитория test_repo



Рисунок 8, созданный репозиторий `test_repo` на GitHub

Переименование репозитория (-R [старое_имя,новое_имя]):

```
(venv) (base) 22:21:52 [cuprumtan@localhost github_API_test]$ python3 cmd.py -R test_repo,test_repo_renamed

No username/password added, using last successfully authorization credentials...

===== USER cuprumtan, TOTAL 14 repos

Successfully renamed repo test_repo to test_repo_renamed
```

Рисунок 9, переименование репозитория



Рисунок 10, переименованный репозиторий `test_repo` в `test_repo_renamed` на GitHub

Удаление репозитория (-D [имя]):


```
(venv) (base) 22:23:52 [cuprumtan@localhost github_API_test]$ python3 cmd.py -D test_repo_renamed

No username/password added, using last successfully authorization credentials...

===== USER cuprumtan, TOTAL 14 repos

Successfully deleted repo test_repo_renamed
```

Рисунок 11, удаление репозитория *test_repo_renamed*

Рекурсивный (--recursively) вывод содержимого репозитория (-T [имя]):

```
(venv) (base) 22:26:18 [cuprumtan@localhost github_API_test]$ python3 cmd.py -T db_import_export --recursively

No username/password added, using last successfully authorization credentials...

===== USER cuprumtan, TOTAL 13 repos

===== REPO db_import_export CONTENTS:
.gitignore
calc_macro
pom.xml
отчет.pdf
csv/doctors.csv
csv/patients.csv
scripts/structurePostgreSQL.sql
scripts/structureSQLite.sql
src/main/java/edu/psu/Main.java
src/main/java/edu/psu/Migration.java
src/main/java/edu/psu/PostgreSQL.java
src/main/java/edu/psu/SQLiteDB.java
```

Рисунок 12, рекурсивный вывод содержимого репозитория *db_import_export*

Поиск файлов определенного типа в репозитории (-F [имя,формат]):

```
(venv) (base) 22:26:38 [cuprumtan@localhost github_API_test]$ python3 cmd.py -F db_import_export,csv

No username/password added, using last successfully authorization credentials...

===== USER cuprumtan, TOTAL 13 repos

===== REPO db_import_export csv FILES:
csv/doctors.csv
csv/patients.csv
```

Рисунок 13, поиск csv-файлов в репозитории *db_import_export*

Изменение видимости репозитория (-P [имя,T|F]):


```
(venv) (base) 22:27:43 [cuprumtan@localhost github API test]$ python3 cmd.py -P db_import_export,T
No username/password added, using last successfully authorization credentials...

===== USER cuprumtan, TOTAL 13 repos

Successfully made repo db_import_export private
```

Рисунок 14, скрытие репозитория `db_import_export`



Рисунок 15, статус `private` репозитория `db_import_export` на `GitHub`

Операции с файлами

Для работы с файлами необходимо указывать название репозитория (`--repo [имя]`). Добавление файла (`-c [имя файла]`) и его заполнение из командной строки (`--content [«текст»]`):

```
(venv) (base) 22:33:46 [cuprumtan@localhost github API test]$ python3 cmd.py --repo db_import_export -c test.txt --content "Hello from command line!"
No username/password added, using last successfully authorization credentials...

===== USER cuprumtan, TOTAL 13 repos

Successfully create file test.txt in repo db_import_export
```

Рисунок 16, добавление файла `test.txt` в репозиторий `db_import_export` и его заполнение из командной строки

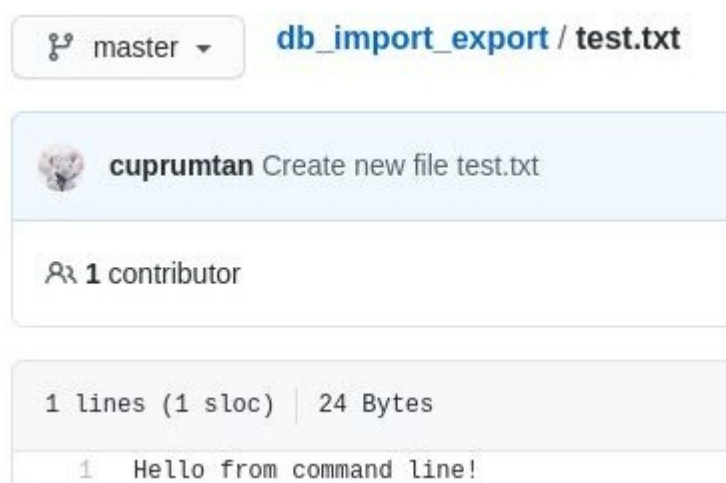


Рисунок 17, отображение файла `test.txt` на `GitHub`

Загрузка существующего файла (-l [имя файла] —path [путь к файлу]):

```
(venv) (base) 22:33:58 [cuprumtan@localhost github_API_test]$ python3 cmd.py --repo db_import_export -l test_2.txt --path /home/cuprumtan/test.txt

No username/password added, using last successfully authorization credentials...

===== USER cuprumtan, TOTAL 13 repos

Successfully loaded file test_2.txt in repo db_import_export
```

Рисунок 18, загрузка существующего файла

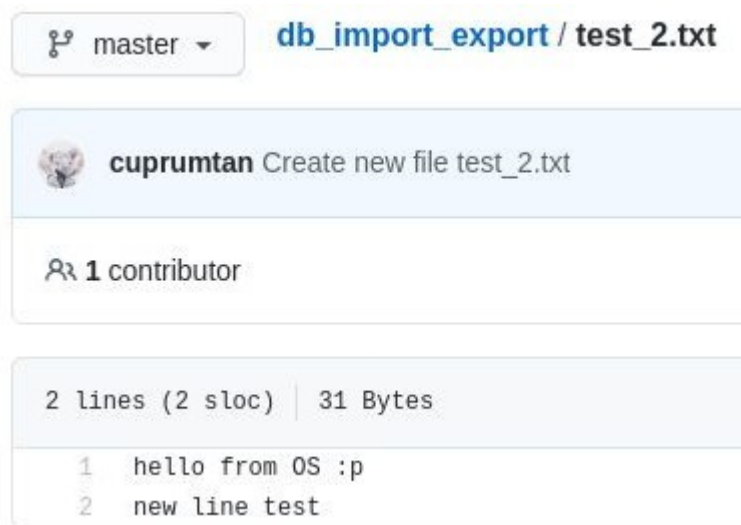


Рисунок 19, отображение файла test_2.txt на GitHub

Удаление файла (-d [файл]):

```
(venv) (base) 22:36:46 [cuprumtan@localhost github_API_test]$ python3 cmd.py --repo db_import_export -d test.txt

No username/password added, using last successfully authorization credentials...

===== USER cuprumtan, TOTAL 13 repos

Successfully deleted file test.txt from repo db_import_export
```

Рисунок 20, удаление файла test.txt

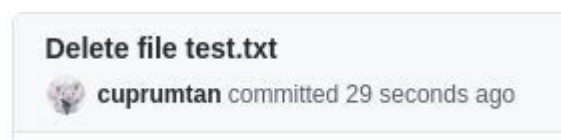


Рисунок 21, коммит на GitHub, говорящий об удалении файла test.txt

Пример содержания лог-файла:

```
2020-10-01 22:23:52.092449 | Renamed repository test_repo to
test_repo_renamed for user cuprumtan
2020-10-01 22:25:11.008812 | Successfully authorized as
cuprumtan
2020-10-01 22:25:12.210416 | Deleted repository
test_repo_renamed for user cuprumtan
2020-10-01 22:26:17.447683 | Successfully authorized as
cuprumtan
2020-10-01 22:26:18.468429 | Loaded repository github_API_test
contents for user cuprumtan
2020-10-01 22:26:33.927993 | Successfully authorized as
cuprumtan
2020-10-01 22:26:38.576796 | Loaded repository db_import_export
contents recursively for user cuprumtan
2020-10-01 22:27:40.624871 | Successfully authorized as
cuprumtan
2020-10-01 22:27:43.518751 | Found csv files in repository
db_import_export for user cuprumtan
2020-10-01 22:28:29.710547 | Successfully authorized as
cuprumtan
2020-10-01 22:28:31.138639 | Setted repository db_import_export
to private for user cuprumtan
2020-10-01 22:29:34.876117 | Successfully authorized as
cuprumtan
2020-10-01 22:29:37.118772 | Setted repository db_import_export
to public for user cuprumtan
2020-10-01 22:33:56.079647 | Successfully authorized as
cuprumtan
2020-10-01 22:33:57.997368 | Created file test.txt in
repository db_import_export for user cuprumtan
2020-10-01 22:35:28.167977 | Successfully authorized as
cuprumtan
2020-10-01 22:35:29.862436 | Loaded file test_2.txt into
repository db_import_export for user cuprumtan
2020-10-01 22:36:38.048276 | Successfully authorized as
cuprumtan
2020-10-01 22:36:40.600296 | Deleted file test_2.txt from
repository db_import_export for user cuprumtan
2020-10-01 22:36:50.051128 | Successfully authorized as
cuprumtan
2020-10-01 22:36:51.869783 | Deleted file test.txt from
repository db_import_export for user cuprumtan
```

Соответствие поставленной задаче

Требование	Статус
Приложение позволяет выполнять	Выполнено полностью. Create-операции:

CRUD-операции над объектами выбранного ресурса	<ul style="list-style-type: none"> ● Создание репозитория; ● Создание файла; Read-операции: <ul style="list-style-type: none"> ● Получение количества всех репозиториях пользователя ● Получение списка всех репозиториях пользователя; ● Получение списка всех файлов в репозитории; Update-операции: <ul style="list-style-type: none"> ● Изменение имени репозитория; ● Изменение видимости репозитория; Delete-операции: <ul style="list-style-type: none"> ● Удаление репозитория; ● Удаление файла;
Приложение позволяет выполнять по крайней мере две CRUD-операции для одного и того же объекта выбранного ресурса	Выполнено полностью.
Приложение выполняет по крайней мере одну из CRUD-операций после успешной аутентификации	Выполнено полностью.
Приложение не требует повторной аутентификации при перезапуске программы	Выполнено полностью.
Параметры успешной аутентификации сохраняются при перезапуске программы в зашифрованном виде	Выполнено полностью.

Список источников

1. <https://github.com/> (дата обращения 01.10.2020).
2. <https://www.python.org/downloads/release/python-374/>,
<https://docs.python.org/3.7/license.html> (дата обращения 01.10.2020).

лицензия

3. https://confluence.jetbrains.com/display/PYH/PyCharm+2019.3.3+Release+Notes?_ga=2.174882918.554375686.1601653569-364471055.1569302137 (дата обращения 01.10.2020).
4. <https://www.jetbrains.com/ru-ru/community/education/#students> (дата обращения 01.10.2020).
5. <https://pypi.org/project/PyGithub/1.53/>,
<https://github.com/PyGithub/PyGithub/blob/master/COPYING> (дата обращения 01.10.2020). лицензия
6. <https://pypi.org/project/PyYAML/5.3.1/>,
<https://github.com/yaml/pyyaml/blob/master/LICENSE> (дата обращения 01.10.2020). лицензия
7. <https://pypi.org/project/cryptography/3.1.1/>,
<https://github.com/pyca/cryptography/blob/master/LICENSE.PSF> (дата обращения 01.10.2020). лицензия
8. <https://pypi.org/project/click/7.1.2/>,
<https://github.com/pallets/click/blob/master/LICENSE.rst> (дата обращения 01.10.2020). лицензия
9. <https://github.com/fernet/spec/> (дата обращения 01.10.2020).