|  |
| --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  федеральное государственное автономное образовательное учреждение высшего образования  «САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ» |

Кафедра компьютерных технологий и программной инженерии

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ОТЧЁТ ПО ПРАКТИКЕ  ЗАЩИЩЁН С ОЦЕНКОЙ  Руководитель |  | | | |
| Ст. преп. |  |  |  | М.Д. Поляк |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

ОТЧЁТ ПО ПРАКТИКЕ

|  |  |  |
| --- | --- | --- |
| вид практики | производственная | |
| тип практики | эксплуатационная | |
| на тему индивидуального задания | | Генерация шаблонов отчётов по лабораторным |
| работам | | | |
|  | | | |

|  |  |
| --- | --- |
| выполнен | Бобрович Николаем Сергеевичем |
| фамилия, имя, отчество обучающегося в творительном падеже | |

|  |  |  |  |
| --- | --- | --- | --- |
| по направлению подготовки | 02.03.03 |  | Математическое обеспечение и |
|  | код |  | наименование направления |
| администрирование информационных систем | | | |
| наименование направления | | | |
| направленности |  |  | Системный анализ в информационных |
|  | код |  | наименование направленности |
| технологиях | | | |
| наименование направленности | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Обучающийся группы № | 4136 |  | Подпись |  | Н.С. Бобрович |
|  | номер |  | подпись, дата |  | инициалы, фамилия |

Санкт–Петербург 2024

1. **Цель работы:**

Разработать скрипт, генерирующий шаблоны отчётов по лабораторным работам.

1. **Исходные данные:**

operating-systems-2024.yam (**Приложение 2**)

1. **Теоретический раздел:**

Создать новую ветку в репозитории. Подготовить PR из этой ветки в main.

На Python + FastAPI реализовать REST эндпоинты:

* GET /courses/{course\_id}/staff - получение списка преподавателей, ведущих предмет
* POST /courses/{course\_id}/groups/{group\_id}/labs/{lab\_id}/template - получение шаблона отчета по лабораторной работе

## GET /courses/{course\_id}/staff

### Входные данные

Отсутствуют

### Логика работы

Эндпоинт возвращает список всех преподавателей, ведущих предмет. Данные берутся из раздела course/staff конфигурационного файла. Подробнее про поиск конфигурационного файла по {course\_id} см. [#1](https://github.com/markpolyak/lab-grader-new/issues/1).

### Выходные данные

Список объектов в формате JSON следующего вида:

[

{

"name": "ФИО преподавателя 1",

"title": "должность преподавателя 1",

"status": "тип занятий, которые ведет преподаватель 1",

},

{

"name": "ФИО преподавателя 2",

"title": "должность преподавателя 2",

"status": "тип занятий, которые ведет преподаватель 2",

},

...

]

## POST /courses/{course\_id}/groups/{group\_id}/labs/{lab\_id}/template

### Входные данные (query string)

Параметр format, который может принимать одно из значений: docx, odf, latex. Например, /courses/{course\_id}/groups/{group\_id}/labs/{lab\_id}/template?format=latex.

### Входные данные (тело запроса)

JSON вида:

{

"github": "Никнейм на GitHub",

"name": "полное ФИО студента (разделитель - пробел), включая дефис вместо отчества, если так указано в личном кабинете ГУАП",

"reviewer": {

"name": "ФИО преподавателя",

"title": "должность преподавателя"

}

}

В теле запроса могут содержаться как оба ключа "github" и "name" одновременно, так и только один из них. Ключ "reviewer" также может отсутствовать, в этом случае ФИО преподавателя и его должность должны быть интерпретированы как пустые строки.

### Логика работы

Найти студента в таблице гугл-документов по его никнейму на GitHub, если он указан (см. более подробное описание в [#1](https://github.com/markpolyak/lab-grader-new/issues/1) ). Если указано ФИО студента - найти его же по ФИО. Если указан и никнейм на GitHub, и ФИО - проверить, что поиск в гугл-таблице по обоим идентификаторам нашел одного и того же студента (одну и ту же строчку на одном и том же листе). Если поиск показал, что ФИО не соответствует никнейму на GitHub, эндпоинт возвращает ошибку 400, а в теле ответа приводится JSON {"message": "Несоответствие ФИО и имени пользователя на GitHub"}.

Сгенерировать шаблон отчета в соответствии с конфигурационным файлом, на который указывают параметры {course\_id} и {lab\_id} (см. [#1](https://github.com/markpolyak/lab-grader-new/issues/1) ). Пример конфигурационного файла см. в [#7](https://github.com/markpolyak/lab-grader-new/issues/7) .

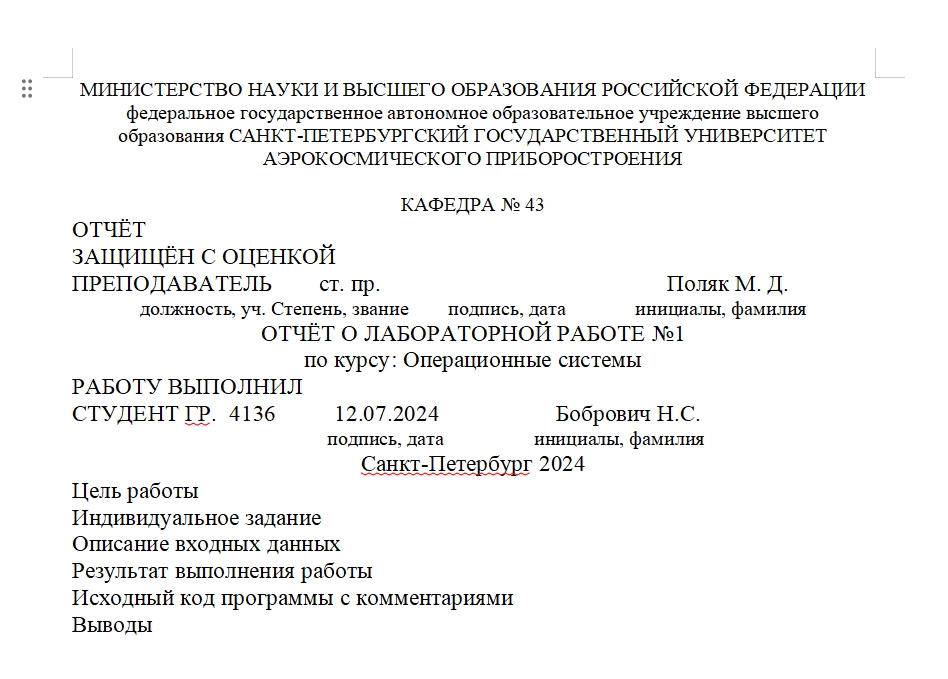
### Выходные данные

При вызове эндпоинта из браузера должен скачиваться соответствующий файл. Для этого необходимо прописать правильные заголовки в ответе, отправляемом сервером на запрос. Тело ответа в этом случае - поток байт (содержимое бинарного или текстового файла с шаблоном отчета).

1. **Практический раздел:**

Разработал скрипт, используя Python и FastAPI (**Приложение 1-2**).

1. **Результат работы:**

Шаблон отчёта формата .odf вида:  


1. **Вывод:**

Разработал скрипт, генерирующий шаблоны отчётов по лабораторным работам. Укрепил свои знания в разработке на Python.

1. **Список литературы:**
2. Таненбаум Э., Бос Х. Современные операционные системы. 2, 3, 4-е изд. 2000- 2018 / Tanenbaum, Andrew. Bos, Herbert. Modern operating systems. 4th ed. 2014
3. Столлингс. Операционные системы. 4-е изд. 2004 / Stallings. Operating systems: internals and design principles. 9th ed. 2017
4. В.Г. Олифер, Н.А. Олифер. Сетевые операционные системы. 2-е изд. 2009
5. Silberschatz et. al. Operating systems concepts. 9th ed. 2012
6. **Приложения:**

**Приложение 1: main.py:**

from fastapi import FastAPI, HTTPException, Path, Query

from pydantic import BaseModel

from typing import List, Optional

import yaml

import os

import gspread

from oauth2client.service\_account import ServiceAccountCredentials

from lab\_template\_generation import generate\_lab\_template

import wget

app = FastAPI()

# Загрузка конфигурационного файла и файла данных

config\_path = 'operating-systems-2024.yaml'

credentials\_path = 'credentials.json'

current\_dir = os.path.dirname(os.path.realpath(\_\_file\_\_))

config\_path = os.path.join(current\_dir, config\_path)

credentials\_path = os.path.join(current\_dir, credentials\_path)

if not os.path.exists(config\_path):

raise FileNotFoundError(f"Configuration file not found: {config\_path}")

with open(config\_path, 'r', encoding='utf-8') as f:

config = yaml.safe\_load(f)

# Проверка наличия ключа 'labs' внутри 'course'

if 'course' in config and 'labs' in config['course']:

labs = config['course']['labs']

else:

raise KeyError("Key 'labs' not found in configuration file")

config\_last\_modified = os.path.getmtime(config\_path)

def reload\_config():

global config, config\_last\_modified

with open(config\_path, 'r', encoding='utf-8') as f:

config = yaml.safe\_load(f)

config\_last\_modified = os.path.getmtime(config\_path)

@app.on\_event("startup")

async def startup\_event():

reload\_config()

@app.middleware("http")

async def check\_config\_update(request, call\_next):

global config, config\_last\_modified

if os.path.getmtime(config\_path) > config\_last\_modified:

reload\_config()

response = await call\_next(request)

return response

# Авторизация в Google Sheets API

scope = ['https://spreadsheets.google.com/feeds', 'https://www.googleapis.com/auth/drive']

creds = ServiceAccountCredentials.from\_json\_keyfile\_name(credentials\_path, scope)

client = gspread.authorize(creds)

sheet = client.open("Student Information").sheet1

# Модель данных для преподавателя

class Teacher(BaseModel):

name: str

title: str

# Модель данных для запроса шаблона отчета

class LabTemplateRequest(BaseModel):

github: Optional[str]

name: Optional[str]

reviewer: Optional[Teacher]

@app.get("/courses/{course\_id}/staff", response\_model=List[Teacher])

async def get\_course\_staff(course\_id: int):

try:

staff = config['course']['staff']

return staff

except KeyError:

raise HTTPException(status\_code=404, detail="Course not found")

@app.post("/courses/{course\_id}/groups/{group\_id}/labs/{lab\_id}/template")

async def get\_lab\_template(

request: LabTemplateRequest,

course\_id: int = Path(..., description="ID курса"),

group\_id: int = Path(..., description="ID группы"),

lab\_id: int = Path(..., description="ID лабораторной работы"),

format: str = Query(..., description="Формат шаблона отчета", choices=["odf"])

):

if request.github:

student = sheet.find(request.github)

elif request.name:

student = sheet.find(request.name)

else:

raise HTTPException(status\_code=400, detail="Either GitHub username or name must be provided")

if student and request.name and request.github:

student\_github = sheet.cell(student.row, 34).value

student\_name = sheet.cell(student.row, 2).value

if student\_github != request.github or student\_name != request.name:

raise HTTPException(status\_code=400, detail="GitHub username and name do not match")

file\_name = f"{' '.join(request.name.split())}\_{group\_id}\_{lab\_id}.odt"

template\_content = generate\_lab\_template(

course\_id=course\_id,

lab\_id=lab\_id,

group\_number=group\_id,

full\_name=request.name.split(),

reviewer\_name=request.reviewer.name if request.reviewer else "",

reviewer\_title=request.reviewer.title if request.reviewer else ""

)

# Сохранение файла и возврат файла в ответе

#with open(file\_name, 'wb') as f:

# f.write(template\_content)

#return {"result": f"{file\_name}"}

return f"<a href='/download/{file\_name}'>Download Lab Template</a>"

#return FileResponse(path=file\_name, filename=file\_name, media\_type='application/vnd.oasis.opendocument.text')

**Приложение 2: lab\_template\_generation.py:**

from typing import List

from odf.opendocument import OpenDocumentText

from odf.text import P

from odf.style import Style, TextProperties, ParagraphProperties

from datetime import datetime

import yaml

import os

from flask import Flask, send\_file

app = Flask(\_\_name\_\_)

@app.route('/download/<filename>')

def download\_file(filename):

return send\_file(filename, as\_attachment=True)

if \_\_name\_\_ == '\_\_main\_\_':

app.run()

# Загрузка конфигурационного файла и файла данных

config\_path = 'operating-systems-2024.yaml'

credentials\_path = 'credentials.json'

current\_dir = os.path.dirname(os.path.realpath(\_\_file\_\_))

config\_path = os.path.join(current\_dir, config\_path)

credentials\_path = os.path.join(current\_dir, credentials\_path)

if not os.path.exists(config\_path):

raise FileNotFoundError(f"Configuration file not found: {config\_path}")

with open(config\_path, 'r', encoding='utf-8') as f:

config = yaml.safe\_load(f)

# Проверка наличия ключа 'labs' внутри 'course'

if 'course' in config and 'labs' in config['course']:

labs = config['course']['labs']

else:

raise KeyError("Key 'labs' not found in configuration file")

def format\_name(name\_parts):

if len(name\_parts) == 3:

return f"{name\_parts[0]} {name\_parts[1][0]}.{name\_parts[2][0]}."

return " ".join(name\_parts)

def generate\_lab\_template(course\_id: int, lab\_id: int, group\_number: int, full\_name: List[str], reviewer\_name: str, reviewer\_title: str) -> bytes:

document = OpenDocumentText()

# Создание стилей

title\_style = Style(name="Title", family="paragraph")

title\_style.addElement(TextProperties(attributes={"fontsize": "12pt", "fontname": "Times New Roman"}))

title\_style.addElement(ParagraphProperties(attributes={"textalign": "center"}))

document.styles.addElement(title\_style)

o\_style = Style(name="O", family="paragraph")

o\_style.addElement(TextProperties(attributes={"fontsize": "14pt", "fontname": "Times New Roman"}))

o\_style.addElement(ParagraphProperties(attributes={"textalign": "center"}))

document.styles.addElement(o\_style)

text\_style = Style(name="Text", family="paragraph")

text\_style.addElement(TextProperties(attributes={"fontsize": "12pt", "fontname": "Times New Roman"}))

text\_style.addElement(ParagraphProperties(attributes={"textalign": "left"}))

document.styles.addElement(text\_style)

u\_style = Style(name="Text", family="paragraph")

u\_style.addElement(TextProperties(attributes={"fontsize": "14pt", "fontname": "Times New Roman"}))

u\_style.addElement(ParagraphProperties(attributes={"textalign": "left"}))

document.styles.addElement(u\_style)

# Добавление текста с определенным шрифтом, размером и ориентацией

title\_paragraph1 = P(stylename=title\_style, text="МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ федеральное государственное автономное образовательное учреждение высшего образования САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ")

document.text.addElement(title\_paragraph1)

# Добавление пустого пространства

empty\_paragraph = P(stylename=title\_style, text=" ")

document.text.addElement(empty\_paragraph)

title\_paragraph1\_4 = P(stylename=title\_style, text="КАФЕДРА № 43")

document.text.addElement(title\_paragraph1\_4)

title\_paragraph2 = P(stylename=text\_style, text="ОТЧЁТ")

document.text.addElement(title\_paragraph2)

title\_paragraph2\_2 = P(stylename=text\_style, text="ЗАЩИЩЁН С ОЦЕНКОЙ")

document.text.addElement(title\_paragraph2\_2)

title\_paragraph2\_3 = P(stylename=text\_style, text=f"ПРЕПОДАВАТЕЛЬ\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0{reviewer\_title}\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0{reviewer\_name}")

document.text.addElement(title\_paragraph2\_3)

title\_paragraph2\_4 = P(stylename=title\_style, text="должность, уч. Степень, звание\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0подпись, дата\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0инициалы, фамилия")

document.text.addElement(title\_paragraph2\_4)

title\_paragraph3 = P(stylename=o\_style, text=f"ОТЧЁТ О ЛАБОРАТОРНОЙ РАБОТЕ №{lab\_id}")

document.text.addElement(title\_paragraph3)

title\_paragraph3\_2 = P(stylename=o\_style, text="по курсу: Операционные системы")

document.text.addElement(title\_paragraph3\_2)

title\_paragraph4 = P(stylename=text\_style, text=f"РАБОТУ ВЫПОЛНИЛ")

document.text.addElement(title\_paragraph4)

title\_paragraph4\_2 = P(stylename=text\_style, text=f"СТУДЕНТ ГР. \u00A0{group\_number}\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0{datetime.now().strftime('%d.%m.%Y')}\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0{format\_name(full\_name)}")

document.text.addElement(title\_paragraph4\_2)

title\_paragraph4\_3 = P(stylename=title\_style, text="\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0подпись, дата\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0\u00A0инициалы, фамилия")

document.text.addElement(title\_paragraph4\_3)

title\_paragraph5 = P(stylename=o\_style, text=f"Санкт-Петербург {datetime.now().year}")

document.text.addElement(title\_paragraph5)

lab\_sections = get\_lab\_content(lab\_id)

if lab\_sections:

for section in lab\_sections:

section\_paragraph = P(stylename=u\_style, text=section)

document.text.addElement(section\_paragraph)

file\_name = f"{' '.join(full\_name)}\_{group\_number}\_{lab\_id}.odt"

document.save(file\_name)

#return f"<a href='/download/{file\_name}'>Download Lab Template</a>"

#return open(file\_name, 'rb').read()

# получение разделов ЛР

def get\_lab\_content(lab\_id: int):

if 'course' not in config or 'labs' not in config['course']:

raise KeyError("Key 'labs' not found in configuration file")

lab\_sections = config['course']['labs'][str(lab\_id)]['report']

return lab\_sections

**Приложение 3: operating-systems-2024.yaml:**

---

# Configuration file for lab-grader

course:

name: Operating systems

alt-names:

- OS

- Операционные системы

- ОС

semester: Spring 2024

email: k43guap@ya.ru

timezone: UTC+3

github:

organization: suai-os-2024

teachers:

- "Mark Polyak"

- markpolyak

google:

spreadsheet: 1xmnbiOoer4wFUozDRya1WLz\_mDIXF2oOEamnKU31EnQ

info-sheet: График

task-id-column: 0

student-name-column: 1

lab-column-offset: 1

staff:

- name: Поляк Марк Дмитриевич

title: ст. преп.

status: лектор

- name: Поляк Марк Дмитриевич

title: ст. преп.

status: лабораторные работы

labs:

"0":

github-prefix: os-task0

short-name: ЛР0

penalty-max: 5

ignore-task-id: True

ci:

- workflows

files:

- goals.md

- info.md

- report.pdf

moss:

language: c

report:

- Задание

- Результат

"01":

github-prefix: os-task-I-1

short-name: ЛР0.1

taskid-max: 20

penalty-max: 6

# ignore-task-id: True

ci:

- workflows

files:

- typescript

moss:

language: c

max-matches: 1000

report:

- Цель работы

- Индивидуальное задание

- Результат выполнения работы

- Содержимое файла typescript c записанной последовательностью команд

- Выводы

"1":

github-prefix: os-task1

short-name: ЛР1

taskid-max: 25

penalty-max: 6

ci:

- workflows

files:

- lab1.sh

moss:

language: c

max-matches: 1000

local-path: lab1

additional:

- suai-os-2020

- suai-os-2021

- suai-os-2022

- suai-os-2023

basefiles:

-

repo: k43guap/os-course-task1

filename: lab1.sh

report:

- Цель работы

- Индивидуальное задание

- Описание входных данных

- Результат выполнения работы

- Исходный код программы с комментариями

- Выводы

"2":

github-prefix: os-task2

short-name: ЛР2

taskid-max: 20

taskid-shift: 4

penalty-max: 9

ci:

- workflows

files:

- lab2.cpp

moss:

language: cc

max-matches: 1000

local-path: lab2

additional:

- suai-os-2020

- suai-os-2021

- suai-os-2022

- suai-os-2023

basefiles:

-

repo: k43guap/os-course-task2

filename: lab2.cpp

-

repo: k43guap/os-course-task2

filename: examples/ex3.cpp

report:

- Цель работы

- Задание на лабораторную работу

- Граф запуска потоков

- Результат выполнения работы

- Исходный код программы с комментариями

- Выводы

"3":

github-prefix: os-task3

short-name: ЛР3

taskid-max: 20

penalty-max: 7

ci:

- workflows

files:

- lab3.cpp

moss:

language: cc

max-matches: 1000

local-path: lab3

additional:

- suai-os-2020

- suai-os-2021

- suai-os-2022

- suai-os-2023

basefiles:

-

repo: k43guap/os-course-task3

filename: lab3.cpp

report:

- Цель работы

- Задание на лабораторную работу

- Граф запуска потоков

- Результат выполнения работы

- Исходный код программы с комментариями

- Выводы

"4":

github-prefix: os-task4

short-name: ЛР4

taskid-max: 30

penalty-max: 8

ci:

- workflows

files:

- lab4.cpp

moss:

language: cc

max-matches: 100

local-path: lab4

additional:

- suai-os-2021

- suai-os-2022

- suai-os-2023

basefiles:

-

repo: k43guap/os-course-task4

filename: lab4.cpp

report:

- Цель работы

- Задание на лабораторную работу

- Описание используемых алгоритмов замещения страниц

- Результат выполнения работы

- Исходный код программы с комментариями

- Выводы

"5":

github-prefix: os-task5

short-name: ЛР5

taskid-max: 30

penalty-max: 10

# ignore-completion-date: True

ci:

workflows:

- run-autograding-tests

- cpplint

files:

- client.cpp

- server.cpp

moss:

language: cc

max-matches: 1000

local-path: lab5

additional:

- suai-os-2021

- suai-os-2022

- suai-os-2023

report:

- Цель работы

- Задание на лабораторную работу

- Схема взаимодействия между клиентом и сервером

- Результат выполнения работы

- Исходный код программы с комментариями

- Выводы

"6":

github-prefix: os-task5

short-name: ЛР6

taskid-max: 30

penalty-max: 7

moss:

language: cc

max-matches: 1000

local-path: lab5

additional:

- suai-os-2021

- suai-os-2022

- suai-os-2023

report:

- Цель работы

- Задание на лабораторную работу

- Описание структуры конфигурационного файла

- Содержимое написанного конфигурационного файла

- Логи сборки проекта в облаке

- Исходный код тестов

- Выводы

misc:

requests-timeout: 5

**Приложение 3: credentials.json:**

{

"type": "service\_account",

"project\_id": "blissful-robot-429110-g5",

"private\_key\_id": "…",

"private\_key": "-----BEGIN PRIVATE KEY-----\n…\n-----END PRIVATE KEY-----\n",

"client\_email": "oskolockkoli@blissful-robot-429110-g5.iam.gserviceaccount.com",

"client\_id": "---",

"auth\_uri": "---",

"token\_uri": "---",

"auth\_provider\_x509\_cert\_url": "https://www.googleapis.com/oauth2/v1/certs",

"client\_x509\_cert\_url": "https://www.googleapis.com/robot/v1/metadata/x509/oskolockkoli%40blissful-robot-429110-g5.iam.gserviceaccount.com",

"universe\_domain": "googleapis.com"

}