

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ
Кафедра интеллектуальных информационных технологий

Отчёт по лабораторной работе №4

Специальность ПО11

Выполнил
И. А. Гурин
студент группы ПО11

Проверил
А. А. Крощенко
ст. преп. кафедры ИИТ,
29.03.2025 г.

Брест 2025

Цель работы: научиться работать с Github API, приобрести практические навыки написания программ для работы с REST API или GraphQL API

Общее задание: используя Github API, реализовать предложенное задание на языке Python. Выполнить визуализацию результатов, с использованием графика или отчета. Можно использовать как REST API (рекомендуется), так и GraphQL

Задание. Анализ активности пользователя на GitHub.

Условие. Напишите Python-скрипт, который:

1. Запрашивает у пользователя имя пользователя GitHub.
2. Получает список публичных репозиторий этого пользователя через GitHub API.
3. Для каждого репозитория получает список последних 10 коммитов и извлекает:
 - Дату коммита
 - Сообщение коммита
 - Автора (если отличается от владельца репозитория)
4. Подсчитывает общее количество коммитов во всех репозиториях.
5. Выводит самые активные месяцы (например, по количеству коммитов).
6. Строит график активности пользователя по месяцам с помощью matplotlib или seaborn.
7. Сохраняет результаты в JSON-файл github_activity.json

Код программы:

main.py:

```
import collections
from datetime import datetime

from api import get_github_data
from json_saver import save_data_to_json
from plot import show_plot

def get_github_username() -> str:
    return input("Enter the github username: ")

def get_user_repositories(data: str) -> str:
    return data["data"]["user"]["repositories"]["edges"]

def print_data(commit_dates, sorted_repos, total_commits):
    print(f"Total commits: {total_commits}")
    formatted_dates = [datetime.strptime(date, "%Y-%m-%dT%H:%M:%SZ").strftime("%B %Y") for date in
                        commit_dates]
    date_counts = collections.Counter(formatted_dates)
    most_active_month, count = date_counts.most_common(1)[0]
    print(f"Most active month: {most_active_month} ({count} commits)")
    print("TOP 3 repositories by number of commits:")
    for i, (repo_name, commit_count) in enumerate(sorted_repos[:3], 1):
        print(f"{i}. {repo_name} ({commit_count} commits)")

def main():
    username: str = get_github_username()
    repositories: str = get_user_repositories(get_github_data(username))
    save_data_to_json(username, repositories)

    total_commits: int = 0
    commit_dates: list = []
    repo_commits: list = []

    for repository in repositories:
        commits_number: int =
int(repository["node"]["defaultBranchRef"]["target"]["history"]["totalCount"])
        total_commits += commits_number
        repo_commits.append((repository["node"]["name"], commits_number))
        for commit in repository["node"]["defaultBranchRef"]["target"]["history"]["edges"]:
            commit_dates.append(commit["node"]["committedDate"])

    sorted_repos = sorted(repo_commits, key=lambda x: x[1], reverse=True)

    print_data(commit_dates, sorted_repos, total_commits)
    show_plot(commit_dates)
```

```
if __name__ == "__main__":
    main()
```

api.py:

```
import requests

from const import GITHUB_API_GRAPHQL_ENDPOINT, GITHUB_TOKEN, QUERY_FILENAME

def get_github_data(username: str) -> str:
    headers = {"Authorization": f"Bearer {GITHUB_TOKEN}"}
    query: str = load_query_from_file(QUERY_FILENAME)
    variables = {"username": username}
    response = requests.post(
        GITHUB_API_GRAPHQL_ENDPOINT, headers=headers, json={"query": query, "variables": variables}
    )
    return response.json()

def load_query_from_file(filename: str) -> str:
    with open(filename, "r", encoding="utf-8") as file:
        return file.read()
```

const.py:

```
import os

GITHUB_TOKEN: str = os.getenv("GITHUB_TOKEN")
GITHUB_API_GRAPHQL_ENDPOINT: str = "https://api.github.com/graphql"
QUERY_FILENAME: str = "query.graphql"
JSON_BASE_FILENAME: str = "repositories.json"
```

json_saver.py:

```
from json import dump

from const import JSON_BASE_FILENAME

def save_data_to_json(username: str, data: str) -> None:
    with open(username + "-" + JSON_BASE_FILENAME, "w", encoding="utf-8") as file:
        dump(data, file, ensure_ascii=False, indent=4)
```

plot.py:

```
from collections import Counter
from datetime import datetime

from matplotlib import pyplot as plt

def show_plot(commit_dates):
    monthly_activity = analyze_commit_activity(commit_dates)
    plot_activity(monthly_activity)

def analyze_commit_activity(commit_dates):
    monthly_activity = Counter()
    for date in commit_dates:
        month = datetime.strptime(date, "%Y-%m-%dT%H:%M:%SZ").strftime("%Y-%m")
        monthly_activity[month] += 1
    return monthly_activity

def plot_activity(monthly_activity):
    sorted_activity = sorted(monthly_activity.items())
    months, counts = zip(*sorted_activity)


    plt.figure(figsize=(10, 6))
    plt.bar(months, counts, color="skyblue")
    plt.xlabel("Month")
    plt.ylabel("Number of commits")
    plt.title("User activity by month")
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()
```

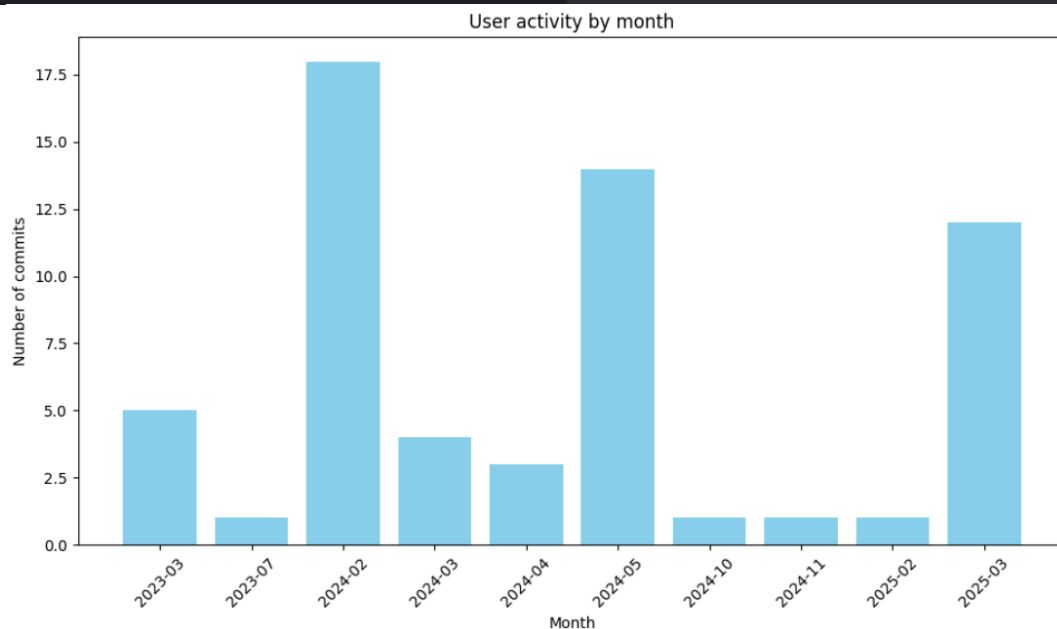
query.graphql:

```
query($username: String!) {
  user(login: $username) {
    repositories(first: 100) {
      edges {
        node {
          name
          defaultBranchRef {
            target {
              ... on Commit {
                history(first: 10) {
                  totalCount
                  edges {
                    node {
                      committedDate
                      message
                      author {
                        name
                      }
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
```

Рисунки с результатами работы программы:

```
Enter the github username: ILGurin
Total commits: 123
Most active month: February 2024 (18 commits)
TOP 3 repositories by number of commits:
1. spp_po11 (52 commits)
2. IIT-University (23 commits)
3. ProxyFlashBot (16 commits)
```

 ILGurin-repositories.json



Вывод: научился работать с Github API, приобрёл практические навыки написания программ для работы с GraphQL API