

Сборщик статистики репозиториев

Итерация 2

Поставленные задачи

- Создать архитектуру “обёртки” над PyGithub и Forgejo
- Создать Dockerfile для разворачивания Forgejo
- Создать интерфейс для “обёртки” над PyGithub
- Реализовать интерфейсы “обёртки” под PyGithub
- Заменить все вызовы PyGithub на вызовы “обёртки”

Результаты

[illegible]

- Создана обёртка
- Обёртка реализована под PyGithub и частично под PyForgejo
- Все вызовы библиотеки PyGithub заменены на вызовы обёртки
- Forgejo разворачивается в Docker-контейнере

Результаты

API можно выбрать, изменив источник и указав базовый URL
(для Forgejo)

```
try:
    clients = git_logger.GitClients("forgejo", tokens, "https://codeberg.org/api/v1")
except Exception as e:
    print(e)
    print(traceback.print_exc())
else:
    client = RepositoryFactory.create_api("forgejo", tokens[0], "https://codeberg.org/api/v1")
```



```
try:
    clients = git_logger.GitClients("github", tokens)
except Exception as e:
    print(e)
    print(traceback.print_exc())
else:
    client = RepositoryFactory.create_api("github", tokens[0])
```

Результаты

Не реализованы следующие сценарии использования:

- Получение списка приглашенных пользователей (отсутствует в Forgejo API)
- Получение списка смежных с Issue Pull Request'ов (в Forgejo API отсутствует GraphQL)
- Обработка исключений PyGithub и PyForgejo

План на следующую итерацию

- Создать замену к запросам GraphQL
- Реализовать получение приглашений другим способом
- Создать тесты в CI для проекта
- Оптимизировать методы в реализациях обёртки