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

```
= pyforgejo == pyforgejo == pyforgejo == pyforgejo == pyforgejo == pyforgejo == pyforgejo/issues "HTTP/1.1 200 ok 2025-03-25 17:06:04,328 = INFO = HTTP Request: GET https://codeberg.org/api/vl/repos/harabat/pyforgejo/issues/comments "HTTP/1.2 (repository name: pyforgejo == pyforgejo == 109157; title: "repos/qowner)/repos/contents/ffilepath) yields a union , 'st int\n\n\n' (n/repos/qowner)/repos/contents/ffilepath) no \n\n\nhas a return type of ContentsReponse, but there's also a not dir\n\nThis fails the pydantic type check when the list gets returned \n\n\n\ostarbox not [] Fix this upstream in the forgejo and\n' on the pyforgejo's ide/upstream fix \n\n-1 and \n' on = [] \text{wisustream} in the forgejo and\n' \n' on the pyforgejo side/upstream fix\n'\n-\n' on \n' on \
   ror]:\n' # .\n' print(endgoint.name, endpoint.path)
ndpoint {endpoint.name}, path_parameters. {endpoint.path | ferdpoint.name}, path_parameters. {endpoint.path | ferdpoint.name}, path_parameters {endpoint.path | ferdpoint.name}, path_parameters {endpoint.path | ferdpoint.name}, path_parameters {endpoint.path | ferdpoint.path | ferdpoint.path | ferdpoint.path | ferdpoint.path | ferdpoint.path.parameters.prom.path | ferdpoint.path.parameters.prom.path | ferdpoint.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.flow.prom.path.parameters.prom.path.flow.prom.path.parameters.prom.path.flow.prom.path.parameters.prom.path.flow.prom.path.parameters.prom.path.flow.prom.path.parameters.prom.path.flow.prom.prom.path.parameters.prom.path.flow.prom.path.parameters.prom.path.flow.prom.prom.path.parameters.prom.path.prom.path.parameters.prom.path.prom.path.parameters.prom.path.prom.path.prom.path.parameters.prom.path.prom.path.parameters.prom.path.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.parameters.prom.path.path.parameters.prom.path.path.path.parameters.prom.path.path.path.parameters.prom.
        blutes,\n context=context,\n in value=[{_links::{_git::https://villes./yolnadirjs/summarys.html.pydantic_corce of ContentsResponse [type=model_type, input_value=[{_links::{_git::https://kEADME.md?ref=master}], input_type=list[]\n partial = False\nfrom_attributes = None\nobject = [{_links::{_git::https://villes.to.org/api/vi/repos/packit-vster/README.md', 'encoding': None, ...]\nself = TypeAdapter(ContentsResponse)\nstrict = None\nn/usr/local/lib/pyt
   sel j, or n = [ ] walt for the upstream fixinn--inincdetalls>csummarys-from message from the tests</summary>\n\n^ \n> ibutes,\n^ co of Contents contexts con
```

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

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

План на следующую итерацию

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