



# Итерация №1

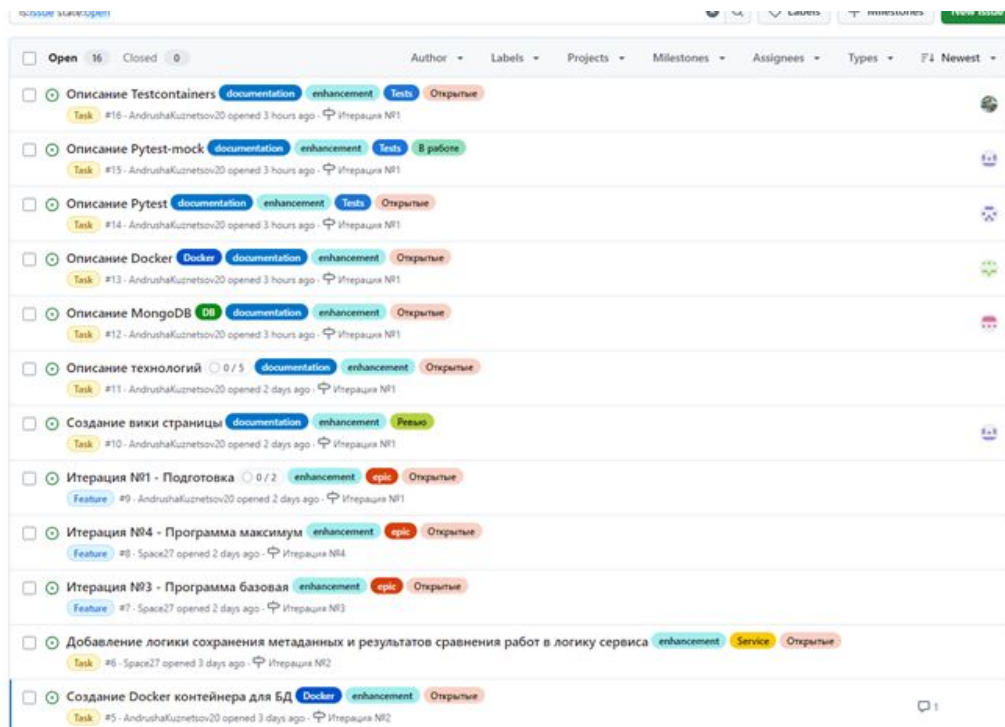
0310 Кузнецов Андрей



# План текущей итерации

- Получить доступ к репозиториям и чатам;
- Провести установочную встречу с заказчиком;
- Подготовить вики-страницу:
  - Описать основную проблему, которую решает проект;
  - Указать категории пользователей, что для каждой из них важно в проекте.
- Создать задачи и фичи; указать теги, версии и описания;
- Определиться с БД, которая будет использоваться на проекте;
- Описать используемые технологии;
- Осуществить запуск проекта;
- Углубиться в кодовую базу.

# Создание задач



The screenshot displays a Jira issue tracker interface. At the top, there are tabs for 'Issues', 'Backlog', 'Reports', and 'Dashboards'. Below the tabs, a filter bar shows 'Open: 16' and 'Closed: 0'. The main list of issues is as follows:

Issue Type	Issue ID	Summary	Labels	Status	Assignee	Created
Task	#16	Описание Testcontainers	documentation, enhancement, Tests	Открытые	AndrushaKuznetsov20	3 hours ago
Task	#15	Описание Pytest-mock	documentation, enhancement, Tests, 8 pafone	Открытые	AndrushaKuznetsov20	3 hours ago
Task	#14	Описание Pytest	documentation, enhancement, Tests	Открытые	AndrushaKuznetsov20	3 hours ago
Task	#13	Описание Docker	Docker, documentation, enhancement	Открытые	AndrushaKuznetsov20	3 hours ago
Task	#12	Описание MongoDB	DB, documentation, enhancement	Открытые	AndrushaKuznetsov20	3 hours ago
Task	#11	Описание технологий	0 / 5, documentation, enhancement	Открытые	AndrushaKuznetsov20	2 days ago
Task	#10	Создание вики страницы	documentation, enhancement, Pafone	Открытые	AndrushaKuznetsov20	2 days ago
Feature	#9	Итерация №1 - Подготовка	0 / 2, enhancement, epic	Открытые	AndrushaKuznetsov20	2 days ago
Feature	#8	Итерация №4 - Программа максимум	enhancement, epic	Открытые	Space27	2 days ago
Feature	#7	Итерация №3 - Программа базовая	enhancement, epic	Открытые	Space27	2 days ago
Task	#6	Добавление логики сохранения метаданных и результатов сравнения работ в логику сервиса	enhancement, Service	Открытые	Space27	3 days ago
Task	#5	Создание Docker контейнера для БД	Docker, enhancement	Открытые	AndrushaKuznetsov20	3 days ago

# Демонстрация доски проекта

The screenshot displays a project management interface for a project named 'mse1h2025-cache'. The interface includes a search bar, a filter dropdown, and four columns representing different task statuses: 'No Status', 'Открытые' (Open), 'В работе' (In Progress), and 'Ревью' (Review).

**Project Details:**

- Project Name: mse1h2025-cache
- View: Доска задач (Task Board)
- Filter: Filter by keyword or by field

**Task Columns:**

- No Status (4 tasks):**
  - mse1h2025-cache #12: Описание MongoDB
  - mse1h2025-cache #13: Описание Docker
  - mse1h2025-cache #14: Описание Pytest
  - mse1h2025-cache #16: Описание Testcontainers
- Открытые (Open) (8 tasks):**
  - mse1h2025-cache #2: Итерация №2 - Программа минимума
  - mse1h2025-cache #3: Создание абстракции для работы с кешем
  - mse1h2025-cache #1: Сохранение метаданных и результатов работ в БД
  - mse1h2025-cache #5: Создание Docker контейнера для БД
  - mse1h2025-cache #4: Сериализация ASTFeatures
  - mse1h2025-cache #6: Добавление логики сохранения метаданных и результатов сравнения работ в логику сервиса
  - mse1h2025-cache #7: Итерация №3 - Программа базовая
- В работе (In Progress) (1 task):**
  - mse1h2025-cache #15: Описание Pytest-mock
- Ревью (Review) (1 task):**
  - mse1h2025-cache #10: Создание вики страницы

# Демонстрация вики-страницы

mse1h2025-cache

16 Pull requests Discussions Actions Projects 1 Wiki Security Insights Settings

Описание проекта

Daniil Lokosov edited this page 2 days ago · [3 revisions](#)

Задача

Для [проекта](#) по поиску схожего программного кода требуется написание механизма для сохранения метаданных исходных кодов, а также результатов проверки в БД типа MongoDB или PostgreSQL. На данный момент сохраняются только результаты проверки в csv файл.

Функциональность

- Программа минимум:
  - Сохранять метаданные работ в БД.
  - Сохранять результаты проверок в БД.
  - При этом кроме как сохранения ничего не происходит, т.е. на сохранённые данные в БД ПО не базируется.
  - Контейнер с БД поставить жить отдельно от основного контейнера и настроить docker-compose. Каталог с данными БД превратить в volume.
- Программа базовая:
  - Сохранять метаданные работ в БД. Сохранять результаты проверок в БД.
  - Уметь извлекать из БД метаданные работ, вместо их перечисления;
  - Уметь извлекать из БД результаты сравнения работ, вместо их перечисления.
  - Контейнер с БД поставить жить отдельно от основного контейнера и настроить docker-compose. Каталог с данными БД превратить в volume.
- Программа максимум:
  - Сохранять метаданные работ в БД. Сохранять результаты проверок в БД.

Pages 2

Find a page...

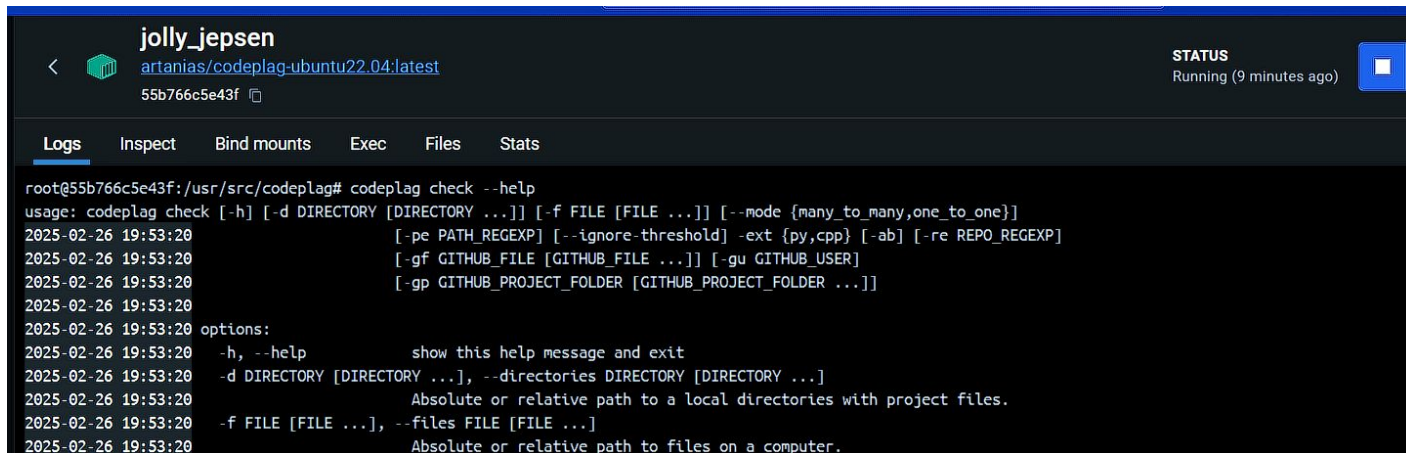
Home

Описание проекта

- Задача
- Функциональнс
- Технологии
- Результат
- Базовый
- Максимальн
- Решаемая проб
- Пользователи
- Сценарии испо
- Обобщенные о
- использования
- Требования

+ Add a cust

# Запуск docker-контейнера



The screenshot shows the Docker Desktop interface for a container named 'jolly\_jepsen'. The container is based on the image 'artanias/codeplag-ubuntu22.04:latest' and has ID '55b766c5e43f'. It is currently running, as indicated by the 'STATUS' section which says 'Running (9 minutes ago)'. The 'Logs' tab is selected, showing the output of the 'codeplag check --help' command. The output includes usage instructions and a list of options.

```
root@55b766c5e43f:/usr/src/codeplag# codeplag check --help
usage: codeplag check [-h] [-d DIRECTORY [DIRECTORY ...]] [-f FILE [FILE ...]] [--mode {many_to_many,one_to_one}]
2025-02-26 19:53:20 [-pe PATH_REGEX] [--ignore-threshold] -ext {py,cpp} [-ab] [-re REPO_REGEX]
2025-02-26 19:53:20 [-gf GITHUB_FILE [GITHUB_FILE ...]] [-gu GITHUB_USER]
2025-02-26 19:53:20 [-gp GITHUB_PROJECT_FOLDER [GITHUB_PROJECT_FOLDER ...]]
2025-02-26 19:53:20
2025-02-26 19:53:20 options:
2025-02-26 19:53:20 -h, --help            show this help message and exit
2025-02-26 19:53:20 -d DIRECTORY [DIRECTORY ...], --directories DIRECTORY [DIRECTORY ...]
2025-02-26 19:53:20                        Absolute or relative path to a local directories with project files.
2025-02-26 19:53:20 -f FILE [FILE ...], --files FILE [FILE ...]
2025-02-26 19:53:20                        Absolute or relative path to files on a computer.
```

# Тестовый прогон

```
root@55b766c5e43f:/usr/src/codeplag# codeplag check --extension py --github-files https://github.com/OSLL/code-plagiarism/blob/main/src/codeplag/pyplag/utlis.py --directories src/codeplag/pyplag/
[WARNING] 16:57 - Env file not found or not a file. Trying to get token from environment.
[WARNING] 16:57 - GitHub access token is not defined.
[INFO] 16:57 - Starting searching for plagiarism ...
+++++
May be similar:
-----
/usr/src/codeplag/src/codeplag/pyplag/utlis.py
2024-10-19 10:57:12
-----
https://github.com/OSLL/code-plagiarism/blob/main/src/codeplag/pyplag/utlis.py
2024-09-20T06:50:18Z
-----

FastMetrics:  JAKKAR  OPERATORS  KEYWORDS  LITERALS  WEIGHTED_AVERAGE
Similarity    100.00%    100.00%    100.00%    100.00%           100.00%

AdditionalMetrics:  Structure
Similarity          100.00%

get_ast_from_content[21]    get_ast_from_content[21]    get_ast_from_filename[63]    get_features_from_ast[94]    _get_works_from_filepaths[102]    PyFeaturesGetter[118]
100.00%                    17.70%                    11.42%                    16.95%                    17.11%
get_ast_from_filename[63]    17.70%                    100.00%                    24.63%                    23.38%                    35.10%
get_features_from_ast[94]    11.42%                    24.63%                    100.00%                    43.04%                    17.05%
_get_works_from_filepaths[102]    16.95%                    23.38%                    43.04%                    100.00%                    22.57%
PyFeaturesGetter[118]    17.11%                    35.10%                    17.05%                    22.57%                    100.00%

+++++
[INFO] 16:57 - Ending searching for plagiarism ...
root@55b766c5e43f:/usr/src/codeplag#
```



## План на следующую итерацию

1. Ссылки на материалы для проверки итерации собрать в README под заголовком «Итерация 2»;
2. Выполнить программу минимум:
  - Сохранять метаданные работ в БД;
  - Сохранять результаты проверок в БД;
  - При этом кроме как сохранения ничего не происходит, т.е. на сохраненные данные в БД ПО не базируется;
  - Контейнер с БД поставить жить отдельно от основного контейнера и настроить docker-compose. Каталог с данными БД превратить в volume.