ANIMEUPDATES BOT

КОМАНДА: ГОРБУНОВА КРИСТИНА, КАРПЕНКО КСЕНИЯ, ПЕТЕЛИН АЛЕКСАНДР

AnimeUpdatesBot —

уведомления о выходе новых эпизодов аниме

Телеграм-бот **AnimeUpdatesBot** позволяет составлять список аниме, которые будут отслеживаться ботом. При выходе нового эпизода бот присылает уведомление с кратким описанием эпизода

Как мы будем это делать

Реализации аналогичных ботов

WEB-SCRAPING

Суровое ручное ковыряние HTML-документов

WEB-SCRAPING 2.0

Использование Request, BeautifulSoup, XPath и других библиотек для web-scraping

AT LAST, API

Реализации таких проектов чрезвычайно редки

Проблема: сложный парсинг и плохой код

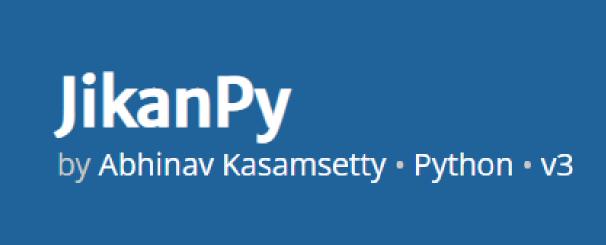
Решение: используем Jikan



Jikan (時間) is an open-source PHP & REST API for the "most active online anime + manga community and database"— MyAnimeList.net. It parses the website to satisfy the need for an API.

JIKAN API

Позволяет получать красивые JSON-ы простыми запросами



JIKANPY

Python Wrapper над Jikan API

Наше MVP



ДОБАВИТЬ АНИМЕ

Добавляет аниме в лист по ссылке на myanimelist



УБРАТЬ АНИМЕ

Удаляет аниме из листа по ссылке на myanimelist



ЛИСТ ПРОСМАТРИВАЕМОГО

Предоставляет доступ к списку добавленных аниме



В**СПОМОГАТЕЛЬНЫЙ** ФУНКЦИОНАЛ

Начинает бота, отображает команды и делает другие must-have вещи

Потенциальные расширения



ПОЛЬЗОВАТЕЛЬСКИЙ ФУНКЦИОНАЛ

- Показ топ-10 сезонного аниме
- Просмотр информации о конкретном аниме по ссылке
- Расписание выхода эпизодов на неделе



ИНТЕРФЕЙС

Добавление кнопок, подсказок, подробного руководства при некорректных действиях пользователя



ВНУТРЕННИЕ ПРОВЕРКИ

Сопоставление данных, получаемых с myanimelist с другой платформой для подтверждения корректности



Расширение функционала на уведомления о новых главах манги

Бонус

Мы не хотим писать так

```
@app.route('/game/<int:game_id>/rounds/<int:round>/category', methods=['GET', 'POST'])
89 def game(game_id, round):
        session = db_session.create_session()
        game_form = GameForm()
        if game_form.validate_on_submit() and game_form.category_submit.data: # если нажата кнопка добавления категории
            if not session.query(Pack).filter((Pack.id == game_id)).first():
                with open(f'games/{game_id}.json', 'w') as f:
                    data = {
                         "rounds": [
                                "categories": [
                    json.dump(data, f, indent=4)
                    pack = Pack(
                        game=f'games/{game_id}.json',
                        user_id=current_user.id
                    ) # в таблицу записывается путь к файлу и id создателя
                    session.add(pack)
                    session.commit()
             category_data = ({
                                 "name": game_form.category.data,
                                 "description": game_form.description.data,
                                 "questions": [
                                         "text": game_form.text.data,
                                         "par": game_form.par.data,
                                         "correct_answers": game_form.answers.data,
                                         "answer_time": game_form.time.data
```